

INFORMATION PROCESSOR AND PROCESSING METHOD, AND RECORDING MEDIUM THEREFOR, AND PROGRAM AND ITS RECORDING MEDIUM

Publication number: JP2002158972 (A)

Publication date: 2002-05-31

Inventor(s): KATO MOTOKI; HAMADA TOSHIYA +

Applicant(s): SONY CORP +

Classification:

- international: *H04N5/85; H04N5/91; H04N5/92; H04N5/93; H04N5/84; H04N5/91; H04N5/92; H04N5/93; (IPC1-7): H04N5/92; H04N5/85; H04N5/91; H04N5/93*

- European:

Application number: JP20010091830 20010328

Priority number(s): JP20010091830 20010328; JP20000183771 20000421; JP20000271552 20000907

Abstract of JP 2002158972 (A)

PROBLEM TO BE SOLVED: To enable to manage commonly AV stream data recorded by analyzing the position of an I picture and AV stream data recorded without analyzing the position of the I picture. **SOLUTION:** CPI-type is described in PlayList(). The CPI type includes EP map type and TU map type. The EP map is used when the position of an I picture can be analyzed, and the TU map is used when the position of the I picture cannot be analyzed.

CPI type	Meaning
0	EP map type
1	TU map type

CPI_type の意味

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2002-158972
(P2002-158972A)

(43) 公開日 平成14年5月31日 (2002.5.31)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)		
H O 4 N	5/92	H O 4 N	5/85	A	5 C 0 5 2
	5/85		5/92	H	5 C 0 5 3
	5/91		5/91	N	
	5/93		5/93	Z	

審査請求 未請求 請求項の数27 O L (全 67 頁)

(21) 出願番号 特願2001-91830 (P2001-91830)
(22) 出願日 平成13年3月28日 (2001.3.28)
(31) 優先権主張番号 特願2000-183771 (P2000-183771)
(32) 優先日 平成12年4月21日 (2000.4.21)
(33) 優先権主張国 日本 (J P)
(31) 優先権主張番号 特願2000-271552 (P2000-271552)
(32) 優先日 平成12年9月7日 (2000.9.7)
(33) 優先権主張国 日本 (J P)

(71) 出願人 000002185
ソニー株式会社
東京都品川区北品川6丁目7番35号
(72) 発明者 加藤 元樹
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内
(72) 発明者 浜田 俊也
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内
(74) 代理人 100082131
弁理士 稲本 義雄

最終頁に続く

(54) 【発明の名称】 情報処理装置および方法、記録媒体、プログラム、並びに記録媒体

(57) 【要約】

【課題】 I ピクチャの位置を分析して記録するAVストリームデータと、分析しないで記録するAVストリームデータとを、共通に管理できるようにする。

【解決手段】 PlayList()には、CPI_typeが記述される。CPI_typeには、EP_maptypeと、TU_map typeがある。I ピクチャの位置が分析できる場合、EP_mapが用いられ、I ピクチャの位置が分析できない場合、TU_mapが用いられる。

CPI_type	Meaning
0	EP_map type
1	TU_map type

CPI_type の意味

【特許請求の範囲】

【請求項1】 AVストリームデータを記録媒体に記録する情報処理装置において、
プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する第1の生成手段と、

記録方法に応じて前記第1のテーブルまたは前記第2のテーブルの一方を選択する選択手段と、

前記選択されたテーブルを前記AVストリームデータとともに前記記録媒体に記録する第1の記録手段とを有することを特徴とする情報処理装置。

【請求項2】 前記第1のテーブルは、EP_mapであり、前記第2のテーブルは、TU_mapであることを特徴とする請求項1に記載の情報処理装置。

【請求項3】 前記選択手段は、ノンコグニザント記録の際には、前記第2のテーブルを選択することを特徴とする請求項1に記載の情報処理装置。

【請求項4】 前記選択手段は、セルフエンコード記録の際には、前記第1のテーブルを選択することを特徴とする請求項1に記載の情報処理装置。

【請求項5】 前記選択手段は、コグニザント記録の際には、前記第1のテーブルを選択することを特徴とする請求項1に記載の情報処理装置。

【請求項6】 前記AVストリームデータの再生を指定する再生指定情報を生成する第2の生成手段と、
前記第2の生成手段により生成された前記再生指定情報を前記記録媒体に記録する第2の記録手段をさらに有し、

前記再生指定情報は、前記AVストリームデータの再生区間の時間情報を、プレゼンテーションタイムベースで表現するか、またはアライバルタイムベースで表現するかを示す種別情報を含むことを特徴とする請求項1に記載の情報処理装置。

【請求項7】 前記AVストリームデータとともに前記第1のテーブルが記録されている場合、前記再生指定情報は、前記AVストリームデータの再生区間の時間情報を、プレゼンテーションタイムベースで表現し、
前記AVストリームデータとともに前記第2のテーブルが記録されている場合、前記再生指定情報は、前記AVストリームデータの再生区間の時間情報を、アライバルタイムベースで表現することを特徴とする請求項6に記載の情報処理装置。

【請求項8】 AVストリームデータを記録媒体に記録する情報処理装置の情報処理方法において、
プレゼンテーションタイムスタンプと、それに対応する

アクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、

記録方法に応じて前記第1のテーブルまたは前記第2のテーブルの一方を選択する選択ステップと、

前記選択されたテーブルを前記AVストリームデータとともに前記記録媒体に記録する記録ステップとを含むことを特徴とする情報処理方法。

【請求項9】 AVストリームデータを記録媒体に記録する情報処理装置のプログラムにおいて、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、

記録方法に応じて前記第1のテーブルまたは前記第2のテーブルの一方を選択する選択ステップと、

前記選択されたテーブルを前記AVストリームデータとともに前記記録媒体に記録する記録ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項10】 AVストリームデータを記録媒体に記録する情報処理装置を制御するコンピュータに、
プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、

記録方法に応じて前記第1のテーブルまたは前記第2のテーブルの一方を選択する選択ステップと、

前記選択されたテーブルを前記AVストリームデータとともに前記記録媒体に記録する記録ステップとを実行させるプログラム。

【請求項11】 記録媒体からAVストリームデータを再生する情報処理装置において、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパ

ットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている前記記録媒体から、前記第1のテーブルまたは前記第2のテーブルの一方を再生する再生手段と、

再生された前記テーブルに基づいて、前記AVストリームデータの出力を制御する制御手段とを有することを特徴とする情報処理装置。

【請求項12】 記録媒体からAVストリームデータを再生する情報処理装置の情報処理方法において、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている前記記録媒体から、前記第1のテーブルまたは前記第2のテーブルの一方を再生する再生ステップと、再生された前記テーブルに基づいて、前記AVストリームデータの出力を制御する制御ステップとを含むことを特徴とする情報処理方法。

【請求項13】 記録媒体からAVストリームデータを再生する情報処理装置のプログラムにおいて、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている前記記録媒体から、前記第1のテーブルまたは前記第2のテーブルの一方を再生する再生ステップと、再生された前記テーブルに基づいて、前記AVストリームデータの出力を制御する制御ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項14】 記録媒体からAVストリームデータを再生する情報処理装置を制御するコンピュータに、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている前記記録媒体から、前記第1のテーブルまたは前記第2のテーブルの一方を再生する再生ステ

ップと、

再生された前記テーブルに基づいて、前記AVストリームデータの出力を制御する制御ステップとを実行させるプログラム。

【請求項15】 AVストリームデータが記録されている記録媒体において、

プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されていることを特徴とする記録媒体。

【請求項16】 AVストリームデータを記録媒体に記録する情報処理装置において、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成手段と、前記AVストリームデータと前記再生指定情報を前記記録媒体に記録する記録手段とを備えることを特徴とする情報処理装置。

【請求項17】 前記副の再生パスは、オーディオデータのアフターレコーディング用のパスであることを特徴とする請求項16に記載の情報処理装置。

【請求項18】 前記第1の情報は、Main_pathであり、

前記第2の情報は、Sub_pathであることを特徴とする請求項16に記載の情報処理装置。

【請求項19】 前記第2の情報は、前記副の再生パスのタイプを表すタイプ情報、前記副の再生パスが参照する前記AVストリームのファイル名、前記副の再生パスの前記AVストリームのイン点とアウト点、および前記再生パスのイン点が、前記主のパスの時間軸上で同期してスタートする前記主のパス上の時刻を含むことを特徴とする請求項16に記載の情報処理装置。

【請求項20】 AVストリームデータを記録媒体に記録する情報処理装置の情報処理方法において、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、前記AVストリームデータと前記再生指定情報を前記記録媒体に記録する記録ステップとを含むことを特徴とする情報処理方法。

【請求項21】 AVストリームデータを記録媒体に記録する情報処理装置のプログラムにおいて、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報によ

り構成される再生指定情報を生成する生成ステップと、前記AVストリームデータと前記再生指定情報を前記記録媒体に記録する記録ステップと

を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項22】 AVストリームデータを記録媒体に記録する情報処理装置を制御するコンピュータに、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、前記AVストリームデータと前記再生指定情報を前記記録媒体に記録する記録ステップとを実行させるプログラム。

【請求項23】 記録媒体からAVストリームデータを再生する情報処理装置において、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生手段と、再生された前記再生指定情報に基づいて、前記AVストリームデータの出力を制御する制御手段とを備えることを特徴とする情報処理装置。

【請求項24】 記録媒体からAVストリームデータを再生する情報処理装置の情報処理方法において、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生ステップと、再生された前記再生指定情報に基づいて、前記AVストリームデータの出力を制御する制御ステップとを含むことを特徴とする情報処理方法。

【請求項25】 記録媒体からAVストリームデータを再生する情報処理装置のプログラムにおいて、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生ステップと、再生された前記再生指定情報に基づいて、前記AVストリームデータの出力を制御する制御ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項26】 記録媒体からAVストリームデータを再生する情報処理装置を制御するコンピュータに、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生ステップと、再生された前記再生指定情報に基づいて、前記AVストリームデータの出力を制御する制御ステップとを実行させるプログラム。

【請求項27】 AVストリームデータが記録されている記録媒体において、

主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報が記録されていることを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は情報処理装置、再生方法、記録媒体、プログラム、並びに記録媒体に関し、特に、GUIなどに説明表示する情報、主の再生経路の情報、副の再生経路の情報、主の再生経路を構成する個々の再生区間の間の接続情報、ユーザが所望したシーンにセットするブックマークやリジューム点の情報などの情報を含むファイルを記録する情報処理装置、再生方法、記録媒体、プログラム、並びに記録媒体に関する。

【0002】

【従来の技術】近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV(Audio Visual)信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するデジタルのAV信号のソース(供給源)としては、CSデジタル衛星放送やBSデジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。

【0003】ここで、これらのソースから供給されるデジタルビデオ信号は、通常MPEG(Moving Picture Experts Group)2方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送由来のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。あるいは、MPEG1 Video、MPEG2 Video、DV方式をはじめとするデジタル記録方式であれば、1度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。

【0004】しかしながら、このような記録方法は、供給されたビットストリームを1度デコードし、その後で帯域制限や再エンコードを行って記録するため、画質の劣化を伴う。画像圧縮されたデジタル信号の記録をする場合、入力されたデジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。ただし、画像圧縮されたデジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録

レートの上限以下になるように、再エンコードをして記録する必要がある。

【0005】また、入力デジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、1度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置が記録媒体の容量をより無駄なく利用できる。

【0006】以上のように、デジタル放送が主流となる将来においては、データストリーマのように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

【0007】

【発明が解決しようとする課題】ところで、上述したような記録装置により記録媒体にAVストリームデータを記録する場合、例えば、高速再生ができるようにするために、Aストリームデータを分析し、Iピクチャの位置を検出して、Iピクチャにアクセスできるようにして記録する場合と、Aストリームデータを分析せず、そのまま記録する場合とがある。

【0008】このような場合、従来、それぞれ専用のアプリケーションプログラムを用意し、それぞれにより、AVストリームを、異なるフォーマットのAVストリーム（高速再生が可能なAVストリーム、または不可能なAVストリーム）として記録媒体に記録するようにしていた。その結果、アプリケーションプログラムの開発に、費用と時間がかかる課題があった。また、それぞれのアプリケーションプログラムにより記録されたAVストリームは、異なるフォーマットのものとなるので、相互の互換性がなくなり、共通の装置で再生することができなくなる課題があった。

【0009】さらに、従来の記録装置では、例えば、オーディオデータを、所謂アフターレコーディングすることが困難である課題があった。

【0010】本発明はこのような状況に鑑みてなされたものであり、その第1の目的は、高速再生が可能なAVストリームと不可能なAVストリームを、共通に管理することができるようにすることにある。

【0011】さらに、第2の目的は、アフターレコーディングを可能にすることにある。

【0012】

【課題を解決するための手段】本発明の第1の情報処理装置は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応

関係を記述する第2のテーブルを生成する第1の生成手段と、記録方法に応じて第1のテーブルまたは第2のテーブルの一方を選択する選択手段と、選択されたテーブルをAVストリームデータとともに記録媒体に記録する第1の記録手段とを有することを特徴とする。

【0013】前記第1のテーブルは、EP_mapであり、第2のテーブルは、TU_mapとすることができる。

【0014】前記選択手段は、ノンコグニザント記録の際には、第2のテーブルを選択することができる。

【0015】前記選択手段は、セルフエンコード記録の際には、第1のテーブルを選択することができる。

【0016】前記選択手段は、コグニザント記録の際には、第1のテーブルを選択することができる。

【0017】前記AVストリームデータの再生を指定する再生指定情報を生成する第2の生成手段と、第2の生成手段により生成された再生指定情報を記録媒体に記録する第2の記録手段をさらに有し、再生指定情報は、AVストリームデータの再生区間の時間情報を、プレゼンテーションタイムベースで表現するか、またはアライバルタイムベースで表現するかを示す種別情報を含むようにすることができる。

【0018】前記AVストリームデータとともに第1のテーブルが記録されている場合、再生指定情報は、AVストリームデータの再生区間の時間情報を、プレゼンテーションタイムベースで表現し、AVストリームデータとともに第2のテーブルが記録されている場合、再生指定情報は、AVストリームデータの再生区間の時間情報を、アライバルタイムベースで表現することができる。

【0019】本発明の第1の情報処理方法は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、記録方法に応じて第1のテーブルまたは第2のテーブルの一方を選択する選択ステップと、選択されたテーブルをAVストリームデータとともに記録媒体に記録する記録ステップとを含むことを特徴とする。

【0020】本発明の第1の記録媒体のプログラムは、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、記録方法に応じて第1のテーブルまたは第2のテーブルの一方を選択する選択ステップと、選択されたテーブルをAVス

トリームデータとともに記録媒体に記録する記録ステップとを含むことを特徴とする。

【0021】本発明の第1のプログラムは、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルを生成する生成ステップと、記録方法に応じて第1のテーブルまたは第2のテーブルの一方を選択する選択ステップと、選択されたテーブルをAVストリームデータとともに記録媒体に記録する記録ステップとを実行させる。

【0022】本発明の第2の情報処理装置は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブルまたは第2のテーブルの一方を再生する再生手段と、再生されたテーブルに基づいて、AVストリームデータの出力を制御する制御手段とを有することを特徴とする。

【0023】本発明の第2の情報処理方法は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブルまたは第2のテーブルの一方を再生する再生ステップと、再生されたテーブルに基づいて、AVストリームデータの出力を制御する制御ステップとを含むことを特徴とする。

【0024】本発明の第2の記録媒体のプログラムは、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブルまたは第2のテーブルの一方を再生する再生ステップと、再生されたテーブルに基づいて、AVストリームデータの出力を制御する制御ステップとを含むことを特徴とする。

【0025】本発明の第2のプログラムは、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、第1のテーブルまたは第2のテーブルの一方を再生する再生ステップと、再生されたテーブルに基づいて、AVストリームデータの出力を制御する制御ステップとを実行させるプログラム。

【0026】本発明の第1の記録媒体は、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットのAVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されていることを特徴とする。

【0027】本発明の第3の情報処理装置は、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成手段と、AVストリームデータと再生指定情報を記録媒体に記録する記録手段とを備えることを特徴とする。

【0028】前記副の再生パスは、オーディオデータのアフターレコーディング用のパスとすることができる。

【0029】前記第1の情報は、Main_pathであり、第2の情報は、Sub_pathとすることができる。

【0030】前記第2の情報は、副の再生パスのタイプを表すタイプ情報、副の再生パスが参照するAVストリームのファイル名、副の再生パスのAVストリームのイン点とアウト点、および再生パスのイン点が、主のパスの時間軸上で同期してスタートする主のパス上の時刻を含むようにすることができる。

【0031】本発明の第3の情報処理方法は、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、AVストリームデータと再生指定情報を記録媒体に記録する記録ステップとを含むことを特徴とする。

【0032】本発明の第3の記録媒体のプログラムは、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、AVストリームデータと再生指定情報を記録媒体に記録する記録ステップとを含むことを特徴とする。

【0033】本発明の第3のプログラムは、主の再生パ

スを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を生成する生成ステップと、AVストリームデータと再生指定情報を記録媒体に記録する記録ステップとを実行させる。

【0034】前記第4の情報処理装置は、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、前記記録媒体から再生する再生手段と、再生された前記再生指定情報に基づいて、前記AVストリームデータの出力を制御する制御手段とを備えることを特徴とする。

【0035】本発明の第4の情報処理方法は、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、記録媒体から再生する再生ステップと、再生された再生指定情報に基づいて、AVストリームデータの出力を制御する制御ステップとを含むことを特徴とする。

【0036】本発明の第4の記録媒体のプログラムは、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、記録媒体から再生する再生ステップと、再生された再生指定情報に基づいて、AVストリームデータの出力を制御する制御ステップとを含むことを特徴とする。

【0037】本発明の第4のプログラムは、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を、記録媒体から再生する再生ステップと、再生された再生指定情報に基づいて、AVストリームデータの出力を制御する制御ステップとを実行させる。

【0038】本発明の第2の記録媒体は、主の再生パスを示す第1の情報と、主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報が記録されていることを特徴とする。

【0039】本発明の第1の情報処理装置および方法、記録媒体のプログラム、プログラム、並びに記録媒体においては、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録される。

【0040】本発明の第2の情報処理装置および方法、記録媒体のプログラム、並びにプログラムにおいては、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレ

スとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体から、そのテーブルが再生され、それに基づいて、出力が制御される。

【0041】本発明の第3の情報処理装置および方法、記録媒体のプログラム、プログラム、並びに第2の記録媒体においては、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報が記録される。

【0042】本発明の第4の情報処理装置および方法、記録媒体のプログラム、並びにプログラムにおいては、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報が記録媒体から再生され、それに基づいて、出力が制御される。

【0043】

【発明の実施の形態】以下に、本発明の実施の形態について、図面を参照して説明する。図1は、本発明を適用した記録再生装置1の内部構成例を示す図である。まず、外部から入力された信号を記録媒体に記録する動作を行う部分の構成について説明する。記録再生装置1は、アナログデータ、または、デジタルデータを入力し、記録することができる構成とされている。

【0044】端子11には、アナログのビデオ信号が、端子12には、アナログのオーディオ信号が、それぞれ入力される。端子11に入力されたビデオ信号は、解析部14とAVエンコーダ15に、それぞれ出力される。端子12に入力されたオーディオ信号は、AVエンコーダ15に出力される。解析部14は、入力されたビデオ信号からシーンチェンジなどの特徴点を抽出する。

【0045】AVエンコーダ15は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、およびAV同期等のシステム情報(S)をマルチプレクサ16に出力する。

【0046】符号化ビデオストリームは、例えば、MPEG(Moving Picture Expert Group)2方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG1方式により符号化されたオーディオストリームや、ドルビーAC3方式により符号化されたオーディオストリーム等である。マルチプレクサ16は、入力されたビデオおよびオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ17を介して多重化ストリーム解析部18とソースパケットタイザ19に出力する。

【0047】多重化ストリームは、例えば、MPEG2トラ

ンスポートストリームやMPEG2プログラムストリームである。ソースパケットタイザ19は、入力された多重化ストリームを、そのストリームを記録させる記録媒体100のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームを符号化する。AVストリームは、ECC（誤り訂正）符号化部20、変調部21で所定の処理が施され、書き込み部22に出力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100にAVストリームファイルを書き込む（記録する）。

【0048】デジタルインタフェースまたはデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子13に輸入される。端子13に輸入されたトランスポートストリームの記録方式には、2通りあり、それらは、トランスペアレントに記録する方式と、記録ビットレートを下げるなどの目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインターフェースとしての端子24から制御部23へ入力される。

【0049】入力トランスポートストリームをトランスペアレントに記録する場合、端子13に輸入されたトランスポートストリームは、多重化ストリーム解析部18と、ソースパケットタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述の入力オーディオ浸透とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0050】入力トランスポートストリームを再エンコードした後に記録する場合、端子13に輸入されたトランスポートストリームは、デマルチプレクサ26に輸入される。デマルチプレクサ26は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)を抽出する。

【0051】デマルチプレクサ26により抽出されたストリーム（情報）のうち、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチプレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ16に出力する。

【0052】一方、デマルチプレクサ26から出力され、マルチプレクサ16に輸入されたオーディオストリームとシステム情報、および、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部18とソースパケットタイザ19にスイッチ17を介して出力される。これ以後の記録媒体100へAVストリームが記録されるまでの処理は、上述の入

力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0053】本実施の形態の記録再生装置1は、AVストリームのファイルを記録媒体100に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部23により作成される。制御部23への入力情報は、解析部14からの動画像の特徴情報、多重化ストリーム解析部18からのAVストリームの特徴情報、および端子24から入力されるユーザからの指示情報である。

【0054】解析部14から供給される動画像の特徴情報は、入力動画像信号の中の特徴的な画像に関する情報であり、例えば、プログラムの開始点、シーンチェンジ点、コマーシャル（CM）の開始・終了点などの指定情報（マーク）であり、また、その指定場所の画像のサムネイル画像の情報も含まれる。

【0055】多重化ストリーム解析部18からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関する情報であり、例えば、AVストリーム内のIピクチャのアドレス情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報、ビデオストリームの中の特徴的な画像に関する情報（マーク）などである。

【0056】端子24からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報などである。

【0057】制御部23は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベース、記録媒体100の記録内容の管理情報(info.dvr)、およびサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様に、ECC符号化部20、変調部21で処理されて、書き込み部22へ入力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100へデータベースファイルを記録する。

【0058】上述したアプリケーションデータベース情報についての詳細は後述する。

【0059】このようにして記録媒体100に記録されたAVストリームファイル（画像データと音声データのファイル）と、アプリケーションデータベース情報が再生される場合、まず、制御部23は、読み出し部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読み出し部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベー

ス情報は、復調部29、ECC復号部30の処理を経て、制御部23へ入力される。

【0060】制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインターフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読み出し部28に指示する。読み出し部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、所定の処理が施されることにより復調され、さらにECC復号部30の処理を経て、ソースデパケッタ31出力される。

【0061】ソースデパケッタ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ26に出力できるストリームに変換する。デマルチプレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、およびAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

【0062】また、ユーザインターフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読み出し部28に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部23は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読み出し部28に指示する。

【0063】また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読み出し部28に指示する。

【0064】読み出し部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

【0065】次に、ユーザが、記録媒体100に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を生

し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインターフェースとしての端子24から再生区間の開始点(イン点)と終了点(アウト点)の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成する。

【0066】ユーザが、記録媒体100に記録されているAVストリームの一部を消去したい場合、ユーザインターフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不要なストリーム部分を消去するように、書き込み部22に指示する。

【0067】ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、かつ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベースを作成し、さらに、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

【0068】まず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読み出し部28にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読み出し部28は、記録媒体100からデータを読み出し、そのデータは、復調部29、ECC復号部30、ソースデパケッタ31を経て、デマルチプレクサ26に出力される。

【0069】制御部23は、デマルチプレクサ26に入力されたデータを解析して、ビデオストリームの再エンコード方法(picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て)と、再多重化方式を決定し、その方式をAVエンコーダ15とマルチプレクサ16に供給する。

【0070】次に、デマルチプレクサ26は、入力されたストリームをビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)に分離する。ビデオストリームは、「AVデコーダ27に入力されるデータ」と「マルチプレクサ16に入力されるデータ」がある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ27で復号され、復号されたピクチャはAVエンコーダ15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ16に入力される。

【0071】マルチプレクサ16は、制御部23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、EC符号化部20、変調部21で処理されて、書き込み部22に入力される。書き込み部22は、制御部23から供給される制御信号に基づいて、記録媒体100にAVストリームを記録する。

【0072】以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図2は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlaylistとClipの2つのレイヤをもつ。Volume Informationは、ディスク内のすべてのClipとPlaylistの管理をする。ここでは、1つのAVストリームとその付属情報のペアを1つのオブジェクトと考え、それをClipと称する。AVストリームファイルはClip AV stream fileと称し、その付属情報は、Clip Information fileと称する。

【0073】1つのClip AV stream fileは、MPEG2トランスポートストリームをアプリケーションフォーマットによって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエン트리ポイントは、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられた時、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

【0074】Playlistについて、図3を参照して説明する。Playlistは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようにするために設けられている。1つのPlaylistは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイン点(IN)とアウト点(OUT)の対で表される。従って、Playlistは、複数のPlayItemが集まることにより構成される。

【0075】Playlistには、2つのタイプがある。1つは、Real Playlistであり、もう1つは、Virtual Playlistである。Real Playlistは、それが参照しているClipのストリーム部分を共有している。すなわち、Real Playlistは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real Playlistが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

【0076】Virtual Playlistは、Clipのデータを共有していない。従って、Virtual Playlistが変更または消去されたとしても、Clipの内容には何も変化が生じない。

【0077】次に、Real Playlistの編集について説明

する。図4(A)は、Real Playlistのクリエイト(create:作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal Playlistが新たに作成される操作である。

【0078】図4(B)は、Real Playlistのディバイド(divide:分割)に関する図であり、Real Playlistが所望な点で分けられて、2つのReal Playlistに分割される操作である。この分割という操作は、例えば、1つのPlaylistにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番組として登録(記録)し直したいといったようなときに行われる。この操作により、Clipの内容が変更される(Clip自体が分割される)ことはない。

【0079】図4(C)は、Real Playlistのコンバイン(combine:結合)に関する図であり、2つのReal Playlistを結合して、1つの新しいReal Playlistにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される(Clip自体が1つにされる)ことはない。

【0080】図5(A)は、Real Playlist全体のデリート(delete:削除)に関する図であり、所定のReal Playlist全体を消去する操作がされた場合、削除されたReal Playlistが参照するClipの、対応するストリーム部分も削除される。

【0081】図5(B)は、Real Playlistの部分的な削除に関する図であり、Real Playlistの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

【0082】図5(C)は、Real Playlistのミニマイズ(Minimize:最小化)に関する図であり、Real Playlistに対応するPlayItemを、Virtual Playlistに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual Playlistにとって不必要なClipの、対応するストリーム部分は削除される。

【0083】上述したような操作により、Real Playlistが変更されて、そのReal Playlistが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual Playlistが存在し、そのVirtual Playlistにおいて、削除されたClipにより問題が生じる可能性がある。

【0084】そのようなことが生じないように、ユーザに、削除という操作に対して、「そのReal Playlistが参照しているClipのストリーム部分を参照しているVirtual Playlistが存在し、もし、そのReal Playlistが消去されると、そのVirtual Playlistもまた消去されることになるが、それでも良いか?」といったメッセージなどを表示させることにより、確認(警告)を促した後、ユーザの指示により削除の処理を実行、または、キ

キャンセルする。または、Virtual Playlistを削除する代わりに、Real Playlistに対してミニマイズの操作が行われるようにする。

【0085】次にVirtual Playlistに対する操作について説明する。Virtual Playlistに対して操作が行われたとしても、Clipの内容が変更されることはない。図6は、アSEMBル(Assemble) 編集 (IN-OUT 編集)に関する図であり、ユーザが見たいと所望した再生区間のPlayItemを作り、Virtual Playlistを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている(後述)。

【0086】図6(A)に示したように、2つのReal Playlist1, 2と、それぞれのRealPlaylistに対応するClip1, 2が存在している場合に、ユーザがReal Playlist1内の所定の区間(In1乃至Out1までの区間: PlayItem1)を再生区間として指示し、続けて再生する区間として、Real Playlist2内の所定の区間(In2乃至Out2までの区間: PlayItem2)を再生区間として指示したとき、図6(B)に示すように、PlayItem1とPlayItem2から構成される1つのVirtual Playlistが作成される。

【0087】次に、Virtual Playlistの再編集(Re-editing)について説明する。再編集には、Virtual Playlistの中のイン点やアウト点の変更、Virtual Playlistへの新しいPlayItemの挿入(insert)や追加(append)、Virtual Playlistの中のPlayItemの削除などがある。また、Virtual Playlistそのものを削除することもできる。

【0088】図7は、Virtual Playlistへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual Playlistへのオーディオのアフレコをサブパスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlaylistのメインパスのAVストリームに、付加的なオーディオストリームが、サブパスとして付加される。

【0089】Real PlaylistとVirtual Playlistで共通の操作として、図8に示すようなPlaylistの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlaylistの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of Playlist(図20などを参照して後述する)によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

【0090】次に、マーク(Mark)について説明する。マークは、ClipおよびPlaylistの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、AVストリームの内容に起因する特徴的なシーンを指定する、例えば、シーンチェンジ点などである。Playlistを再生する時、そのPlaylistが参照するClipのマークを参照して、使用する事ができる。

【0091】Playlistに付加されるマークは、主にユーザによってセットされる、例えば、ブックマークやリジューム点などである。ClipまたはPlaylistにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【0092】次にサムネイルについて説明する。サムネイルは、Volume、Playlist、およびClipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル(不図示)などを操作して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

【0093】Volumeと各Playlistは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク(記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクと記述する)を記録再生装置1の所定の場所にセットした時に、そのディスクの内容を表す静止画を最初に表示する場合などに用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを想定している。

【0094】Playlistの代表画として、Playlistの最初の画像をサムネイル(代表画)にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上2種類のサムネイルをメニューサムネイルと称する。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、すべてのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図10に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

【0095】一方、ClipとPlaylistには、複数のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることが出来るようにする必要がある。このようなマーク点を表すピクチャをマークサムネイル(Mark Thumbnails)と称する。従って、サムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

【0096】図11は、Playlistに付けられるマークと、そのマークサムネイルの関係について示す図であ

り、図12は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表す時に、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置1がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。

【0097】また、ボリューム内に存在するファイル数を減らすために、すべてのマークサムネイルは1つのファイルに格納するのがよい。Playlistはメニューサムネイル1つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

【0098】図13は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、Playlist、およびClipの関係について示した図である。メニューサムネイルファイルには、Playlist毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各Playlist毎と各Clip毎に作成されたサムネイルがファイルされている。

【0099】次に、CPI (Characteristic Point Information) について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられた時、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本実施の形態では、2種類のCPIを用いる。1つは、EP_mapであり、もう1つは、TU_mapである。

【0100】EP_mapは、エン트리ポイント(EP)データのリストであり、それはエレメンタリストリームおよびトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエン트리ポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ(PTS)と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

【0101】EP_mapは、主に2つの目的のために使用される。第1に、Playlistの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができるとき、EP_mapが作成され、ディスクに記録される。

【0102】TU_mapは、デジタルインタフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット(TU)データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU_mapが作成され、ディスクに記録される。

【0103】STCInfoは、MPEG2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れるかもしれない。そのため、AVストリーム上のある時刻をPTSベースで指す場合、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。更に、そのPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットではSTC-sequenceと呼び、そのインデックスをSTC-sequence-idと呼ぶ。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。STC-sequence-idは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

【0104】プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ1つのシステムタイムベースを共有するものである。再生装置(図1の記録再生装置1)にとって、AVストリームのデコードに先だち、そのAVストリームの内容がわかることは有用である。例えば、ビデオやオーディオのエレメンタリストリームを伝送するトランスポートパケットのPIDの値や、ビデオやオーディオのコンポーネント種類(例えば、HDTVのビデオとMPEG-2AACのオーディオストリームなど)などの情報である。この情報はAVストリームを参照するところのPlaylistの内容をユーザに説明するところのメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だち、再生装置のAVデコードおよびデマルチプレクサの初期状態をセットするために役立つ。この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

【0105】MPEG2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリストリームを伝送するところのトランスポートパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化するなどである。

【0106】ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定める

ところのプログラム内容が一定である区間をProgram-sequenceと呼ぶ。Program-sequenceは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

【0107】本実施の形態では、セルフエンコードのストリームフォーマット (SESF) を定義する。SESFは、アナログ入力信号を符号化する目的、およびデジタル入力信号 (例えばDV) をデコードしてからMPEG2トランスポートストリームに符号化する場合に用いられる。

【0108】SESFは、MPEG-2トランスポートストリームおよびAVストリームについてのエレメンタリーストリームの符号化制限を定義する。記録再生装置1が、SESFストリームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

【0109】デジタル放送のストリームは、次に示す方式のうちのいずれかが用いられて記録媒体100に記録される。まず、デジタル放送のストリームをSESFストリームにトランスコーディングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【0110】あるいは、デジタル放送ストリームを構成するエレメンタリーストリームを新しいエレメンタリーストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【0111】例えば、入力ストリームがISDB (日本のデジタルBS放送の規格名称) 準拠のMPEG-2トランスポートストリームであり、それがHDTVビデオストリームとMPEG AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスコーディングし、そのSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトランスポートストリームは、共にISDBフォーマットに準拠しなければならない。

【0112】デジタル放送のストリームが、記録媒体100に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する (入力トランスポートストリームを何も変更しないで記録する) 場合であり、その時にEP_mapが作成されてディスクに記録される。

【0113】または、入力トランスポートストリームをトランスペアレントに記録する (入力トランスポートストリームを何も変更しないで記録する) 場合であり、その時にTU_mapが作成されてディスクに記録される。

【0114】次にディレクトリとファイルについて説明する。以下、記録再生装置1をDVR (Digital Video Recording) と適宜記述する。図14はディスク上のディレ

クトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図14に示したように、“DVR”ディレクトリを含むrootディレクトリ、“PLAYLIST”ディレクトリ、“CLIPINF”ディレクトリ、“M2TS”ディレクトリ、および“DATA”ディレクトリを含む“DVR”ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしても良いが、それらは、本実施の形態のアプリケーションフォーマットでは、無視されるとする。

【0115】“DVR”ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。“DVR”ディレクトリは、4個のディレクトリを含む。“PLAYLIST”ディレクトリの下には、Real PlaylistとVirtual Playlistのデータベースファイルが置かれる。このディレクトリは、Playlistが1つもなくても存在する。

【0116】“CLIPINF”ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。“M2TS”ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。“DATA”ディレクトリは、デジタルTV放送などのデータ放送のファイルがストアされる。

【0117】“DVR”ディレクトリは、次に示すファイルをストアする。“info.dvr”ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ1つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されるとする。“menu.thmb”ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは1つのメニューサムネイルがなければならない。ファイル名は、menu.thmbに固定されるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくても良い。

【0118】“mark.thmb”ファイルは、マークサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは1つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくても良い。

【0119】“PLAYLIST”ディレクトリは、2種類のPlaylistファイルをストアするものであり、それらは、Real PlaylistとVirtual Playlistである。“xxxxx.rpls”ファイルは、1つのReal Playlistに関連する情報をストアする。それぞれのReal Playlist毎に、1つのファイルが作られる。ファイル名は、“xxxxx.rpls”である。ここで、“xxxxx”は、5個の0乃至9まで数字である。ファイル拡張子は、“rpls”でなければならないとする。

【0120】“yyyyy.vpls”ファイルは、1つのVirtual Playlistに関連する情報をストアする。それぞれのVirt

ual Playlist毎に、1つのファイルが作られる。ファイル名は、"yyyy.vpls"である。ここで、"yyyy"は、5個の0乃至9まで数字である。ファイル拡張子は、"vpls"でなければならないとする。

【0121】"CLIPINF"ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。"zzzzz.clpi"ファイルは、1つのAVストリームファイル(Clip AV stream file または Bridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、"zzzzz.clpi"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"clpi"でなければならないとする。

【0122】"M2TS"ディレクトリは、AVストリームのファイルをストアする。"zzzzz.m2ts"ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream fileまたはBridge-Clip AV streamである。ファイル名は、"zzzzz.m2ts"であり、"zzzzz"は、5個の0乃至9までの数字である。ファイル拡張子は、"m2ts"でなければならないとする。

【0123】"DATA"ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイルなどである。

【0124】次に、各ディレクトリ(ファイル)のシンタクスとセマンティクスを説明する。まず、"info.dvr"ファイルについて説明する。図15は、"info.dvr"ファイルのシンタクスを示す図である。"info.dvr"ファイルは、3個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、およびMakerPrivateData()である。

【0125】図15に示したinfo.dvrのシンタクスについて説明するに、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0126】MakerPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word(パディングワード)は、info.dvrのシンタクスに従って挿入される。N1とN2は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【0127】DVRVolume()は、ボリューム(ディスク)の内容を記述する情報をストアする。図16は、DVRVolume()のシンタクスを示す図である。図16に示したDVRVolume()のシンタクスを説明するに、version_numberは、このDVRVolume()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化される。

【0128】lengthは、このlengthフィールドの直後か

らDVRVolume()の最後までDVRVolume()のバイト数を示す32ビットの符号なし整数で表される。

【0129】ResumeVolume()は、ボリュームの中で最後に再生したReal PlaylistまたはVirtual Playlistのファイル名を記憶している。ただし、Real PlaylistまたはVirtual Playlistの再生をユーザが中断した時の再生位置は、PlaylistMark()において定義されるresume-markにストアされる。

【0130】図17は、ResumeVolume()のシンタクスを示す図である。図17に示したResumeVolume()のシンタクスを説明するに、valid_flagは、この1ビットのフラグが1にセットされている場合、resume_PlayList_nameフィールドが有効であることを示し、このフラグが0にセットされている場合、resume_PlayList_nameフィールドが無効であることを示す。

【0131】resume_PlayList_nameの10バイトのフィールドは、リジュームされるべきReal PlaylistまたはVirtual Playlistのファイル名を示す。

【0132】図16に示したDVRVolume()のシンタクスのなかの、UIAppInfoVolumeは、ボリュームについてのユーザインターフェースアプリケーションのパラメータをストアする。図18は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティクスを説明するに、character_setの8ビットのフィールドは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

【0133】name_lengthの8ビットフィールドは、Volume_nameフィールドの中に示されるボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0134】Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号(パスワード)を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せる事(再生される事)が許可される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せる事が許可される。

【0135】最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが0にセットされているか、または、このフラグが1にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置1は、そのディスクの中のPlaylistの一覧を表示させる。それぞれのPlaylistの再生制限は、volume_p

rotect_flagとは無関係であり、それはUIAppInfoPlayList()の中に定義されるplayback_control_flagによって示される。

【0136】PINは、4個の0乃至9までの数字で構成され、それぞれの数字は、ISO/IEC 646に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

【0137】次に図15に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、Playlist(Real PlaylistとVirtual Playlist)のファイル名をストアする。ボリュームに記録されているすべてのPlaylistファイルは、TableOfPlayLists()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlaylistのデフォルトの再生順序を示す。

【0138】図20は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明するに、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバーを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0139】lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す32ビットの符号なしの整数である。number_of_PlayListsの16ビットのフィールドは、Playlist_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlaylistの数に等しくなければならない。Playlist_file_nameの10バイトの数字は、Playlistのファイル名を示す。

【0140】図21は、TableOfPlayLists()のシンタクスを別実施の構成を示す図である。図21に示したシンタクスは、図20に示したシンタクスに、UIAppInfoPlaylist(後述)を含ませた構成とされている。このように、UIAppInfoPlaylistを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図20に示したシンタクスを用いるとして以下の説明をする。

【0141】図15に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置1のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、それを定義し

たメーカーを識別するために標準化されたmaker_IDを持つ。MakersPrivateData()は、1つ以上のmaker_IDを含んでも良い。

【0142】所定のメーカーが、プライベートデータを挿入したい時に、すでに他のメーカーのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカーは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータをMakersPrivateData()の中に追加するようにする。このように、本実施の形態においては、複数のメーカーのプライベートデータが、1つのMakersPrivateData()に含まれることが可能であるようにする。

【0143】図22は、MakersPrivateDataのシンタクスを示す図である。図22に示したMakersPrivateDataのシンタクスについて説明するに、version_numberは、このMakersPrivateData()のバージョンナンバーを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData()の最後までのMakersPrivateData()のバイト数を示す32ビットの符号なし整数を示す。

【0144】mpd_blocks_start_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd_block()の先頭バイトアドレスを示す。相対バイト数はゼロからカウントされる。number_of_maker_entriesは、MakersPrivateData()の中に含まれているメーカープライベートデータのエン트리数を与える16ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker_IDの値を持つメーカープライベートデータが2個以上存在してはならない。

【0145】mpd_block_sizeは、1024バイトを単位として、1つのmpd_blockの大きさを与える16ビットの符号なし整数である。例えば、mpd_block_size=1ならば、それは1つのmpd_blockの大きさが1024バイトであることを示す。number_of_mpd_blocksは、MakersPrivateData()の中に含まれるmpd_blockの数を与える16ビットの符号なし整数である。maker_IDは、そのメーカープライベートデータを作成したDVRシステムの製造メーカーを示す16ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンスによって指定される。

【0146】maker_model_codeは、そのメーカープライベートデータを作成したDVRシステムのモデルナンバーコードを示す16ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカーによって設定される。start_mpd_block_numberは、そのメーカープライベートデータが開始されるmpd_blockの番号を示す16ビットの符号なし整数である。メーカープライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない

い。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

【0147】mpd_lengthは、バイト単位でメーカプライベートデータの大きさを示す32ビットの符号なし整数である。mpd_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中のすべてのmpd_blockは、同じサイズでなければならない。

【0148】次に、Real Playlist fileとVirtual Playlist fileについて、換言すれば、xxxxx.rplsとyyyyy.vplsについて説明する。図23は、xxxxx.rpls (Real Playlist)、または、yyyyy.vpls (Virtual Playlist) のシンタックスを示す図である。xxxxx.rplsとyyyyy.vplsは、同一のシンタックス構成をもつ。xxxxx.rplsとyyyyy.vplsは、それぞれ、3個のオブジェクトから構成され、それらは、Playlist()、PlaylistMark()、およびMakerPrivateData()である。

【0149】PlaylistMark_Start_addressは、Playlistファイルの先頭のバイトからの相対バイト数を単位として、PlaylistMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0150】MakerPrivateData_Start_addressは、Playlistファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0151】padding_word (パディングワード) は、Playlistファイルのシンタックスにしたがって挿入され、N1とN2は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【0152】ここで、既に、簡便に説明したが、Playlistについてさらに説明する。ディスク内にあるすべてのReal Playlistによって、Bridge-Clip (後述) を除くすべてのClipの中の再生区間が参照されていないから。かつ、2つ以上のReal Playlistが、それらのPlaylistで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

【0153】図24を参照してさらに説明するに、図24(A)に示したように、全てのClipは、対応するReal Playlistが存在する。この規則は、図24(B)に示したように、編集作業が行われた後においても守られる。従って、全てのClipは、どれかしらのReal Playlistを参照することにより、必ず視聴することが可能である。

【0154】図24(C)に示したように、Virtual Playlistの再生区間は、Real Playlistの再生区間またはBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual Playlistにも参照されないBridge-Clipがディスクの中に存在してはならない。

【0155】RealPlaylistは、Playlistのリストを含むが、SubPlaylistを含んではならない。Virtual Playlist

は、Playlistのリストを含み、Playlist()の中に示されるCPI_typeがEP_map typeであり、かつPlaylist_typeが0 (ビデオとオーディオを含むPlaylist) である場合、Virtual Playlistは、ひとつのSubPlaylistを含む事ができる。本実施の形態におけるPlaylist()では、SubPlaylistはオーディオのアフレコの目的にだけに使用される、そして、1つのVirtual Playlistが持つSubPlaylistの数は、0または1でなければならない。

【0156】次に、Playlistについて説明する。図25は、Playlistのシンタックスを示す図である。図25に示したPlaylistのシンタックスを説明するに、version_numberは、このPlaylist()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からPlaylist()の最後までまでのPlaylist()のバイト数を示す32ビットの符号なし整数である。Playlist_typeは、このPlaylistのタイプを示す8ビットのフィールドであり、その一例を図26に示す。

【0157】CPI_typeは、1ビットのフラグであり、Playlist()およびSubPlaylist()によって参照されるClipのCPI_typeの値を示す。1つのPlaylistによって参照される全てのClipは、それらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、Playlistの中にあるPlaylistの数を示す16ビットのフィールドである。

【0158】所定のPlaylist()に対応するPlaylist_idは、Playlist()を含むfor-loopの中で、そのPlaylist()の現れる順番により定義される。Playlist_idは、0から開始される。number_of_SubPlayItemsは、Playlistの中にあるSubPlaylistの数を示す16ビットのフィールドである。この値は、0または1である。付加的なオーディオストリームのパス (オーディオストリームパス) は、サブパスの一種である。

【0159】次に、図25に示したPlaylistのシンタックスのUIAppInfoPlaylistについて説明する。UIAppInfoPlaylistは、Playlistについてのユーザインターフェースアプリケーションのパラメータをストアする。図27は、UIAppInfoPlaylistのシンタックスを示す図である。図27に示したUIAppInfoPlaylistのシンタックスを説明するに、character_setは、8ビットのフィールドであり、Playlist_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示したテーブルに準拠する値に対応する。

【0160】name_lengthは、8ビットフィールドであり、Playlist_nameフィールドの中に示されるPlaylist名のバイト長を示す。Playlist_nameのフィールドは、Playlistの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはPlaylistの名称を示す。Playlist_nameフィ

ールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0161】record_time_and_dateは、Playlistが記録された時の日時をストアする56ビットのフィールドである。このフィールドは、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、2001/12/23:01:02:03 は、“0x20011223010203”と符号化される。

【0162】durationは、Playlistの総再生時間を時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary CodedDecimal (BCD)で符号化したものである。例えば、01:45:30は、“0x014530”と符号化される。

【0163】valid_periodは、Playlistが有効である期間を示す32ビットのフィールドである。このフィールドは、8個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、記録再生装置1は、この有効期間の過ぎたPlaylistを自動消去する、といったように用いられる。例えば、2001/05/07 は、“0x20010507”と符号化される。

【0164】maker_idは、そのPlaylistを最後に更新したDVRプレーヤ（記録再生装置1）の製造者を示す16ビットの符号なし整数である。maker_idに符号化される値は、DVRフォーマットのライセンスによって割り当てられる。maker_codeは、そのPlaylistを最後に更新したDVRプレーヤのモデル番号を示す16ビットの符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

【0165】playback_control_flagのフラグが1にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlaylistは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlaylistを視聴することができる。

【0166】write_protect_flagは、図28（A）にテーブルを示すように、1にセットされている場合、write_protect_flagを除いて、そのPlaylistの内容は、消去および変更されない。このフラグが0にセットされている場合、ユーザは、そのPlaylistを自由に消去および変更できる。このフラグが1にセットされている場合、ユーザが、そのPlaylistを消去、編集、または上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

【0167】write_protect_flagが0にセットされているReal Playlistが存在し、かつ、そのReal PlaylistのClipを参照するVirtual Playlistが存在し、そのVirtual Playlistのwrite_protect_flagが1にセットされている場合、ユーザが、RealPlaylistを消去しようとする場合、記録再生装置1は、そのReal Playlistを消去す

る前に、上記Virtual Playlistの存在をユーザに警告するか、または、そのReal Playlistを“Minimize”する。

【0168】is_played_flagは、図28（B）に示すように、フラグが1にセットされている場合、そのPlaylistは、記録されてから一度は再生されたことを示し、0にセットされている場合、そのPlaylistは、記録されてから一度も再生されたことがないことを示す。

【0169】archiveは、図28（C）に示すように、そのPlaylistがオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。ref_thumbnail_index のフィールドは、Playlistを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlaylistには、Playlistを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thum ファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのPlaylistには、Playlistを代表するサムネイル画像が付加されていない。

【0170】次にPlayItemについて説明する。1つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_information_file_name、Clipの再生区間を特定するためのIN_timeとOUT_timeのペア、Playlist()において定義されるCPI_typeがEP_map typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、および、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection_conditionである。

【0171】Playlistが2つ以上のPlayItemから構成される時、それらのPlayItemはPlaylistのグローバル時間軸上に、時間のギャップまたはオーバーラップなしに一系列に並べられる。Playlist()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持たない時、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じSTC連続区間上の時間を指していないなければならない。そのような例を図29に示す。

【0172】図30は、Playlist()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持つ時、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time（図の中でIN_time1と示されているもの）は、先行するPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT_time（図の中でOUT_time1と示されているもの）は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

【0173】現在のPlayItemのIN_time (図の中でIN_time2と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのPlayItemのOUT_time (図の中でOUT_time2と示されているもの)は、現在のPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。

【0174】図31に示すように、PlayList()のCPI_typeがTU_map typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AVストリーム上の時間を指している。

【0175】PlayItemのシンタクスは、図32に示すようになる。図32に示したPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、ClipInformation fileのファイル名を示す。このClipInformation fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

【0176】STC_sequence_idは、8ビットのフィールドであり、PlayItemが参照するSTC連続区間のSTC_sequence_idを示す。PlayList()の中で指定されるCPI_typeがTU_map typeである場合、この8ビットフィールドは何も意味を持たず、0にセットされる。IN_timeは、32ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図33に示すように、PlayList()において定義されるCPI_typeによって異なる。

【0177】OUT_timeは、32ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図34に示すように、PlayList()において定義されるCPI_typeによって異なる。

【0178】Connection_Conditionは、図35に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す2ビットのフィールドである。図36は、図35に示したConnection_Conditionの各状態について説明する図である。

【0179】次に、BridgeSequenceInfoについて、図37を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information fileを指定するBridge_Clip_Information_file_nameを含む。

【0180】また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN_exit_from_previous_Clipと称される。さらに現在のPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットの前にBr

idge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN_enter_to_current_Clipと称される。

【0181】図37において、RSPN_arrival_time_discontinuityは、the Bridge-Clip AVstreamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、ClipInfo()の中において定義される。

【0182】図38は、BridgeSequenceInfoのシンタクスを示す図である。図38に示したBridgeSequenceInfoのシンタクスを説明するに、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、'Bridge-Clip AV stream'を示していなければならない。

【0183】RSPN_exit_from_previous_Clipの32ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【0184】RSPN_enter_to_current_Clipの32ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【0185】次に、SubPlayItemについて、図39を参照して説明する。SubPlayItem()の使用は、PlayList()のCPI_typeがEP_map typeである場合にだけに許される。本実施の形態においては、SubPlayItemはオーディオのアフレコの目的のためだけに使用されるとする。SubPlayItem()は、次に示すデータを含む。まず、PlayListの中のsub_pathが参照するClipを指定するためのClip_information_file_nameを含む。

【0186】また、Clipの中のsub_pathの再生区間を指定するためのSubPath_IN_time と SubPath_OUT_timeを含む。さらに、main pathの時間軸上でsub_pathが再生開始する時刻を指定するためのsync_PlayItem_id と sync_start_PTS_of_PlayItemを含む。sub_pathに参照されるオーディオのClip AV streamは、STC不連続点(シス

テムタイムベースの不連続点)を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

【0187】図40は、SubPlayItemのシンタクスを示す図である。図40に示したSubPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示し、それはPlayListの中でsub pathによって使用される。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

【0188】SubPath_typeの8ビットのフィールドは、sub pathのタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

【0189】sync_PlayItem_idの8ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻が含まれるPlayItemのPlayItem_idを示す。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される(図25参照)。

【0190】sync_start_PTS_of_PlayItemの32ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻を示し、sync_PlayItem_idで参照されるPlayItem上のPTS(Presentation Time Stamp)の上位32ビットを示す。SubPath_IN_timeの32ビットフィールドは、Sub pathの再生開始時刻をストアする。SubPath_IN_timeは、Sub Pathの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

【0191】SubPath_OUT_timeの32ビットフィールドは、Sub pathの再生終了時刻をストアする。SubPath_OUT_timeは、次式によって算出されるPresentation_end_TSの値の上位32ビットを示す。 $Presentation_end_TS = PTS_out + AU_duration$ ここで、PTS_outは、SubPathの最後のプレゼンテーションユニットに対応する33ビット長のPTSである。AU_durationは、SubPathの最後のプレゼンテーションユニットの90kHz単位の表示期間である。

【0192】次に、図23に示したxxxxx.rplsとyyyyy.vplsのシンタクス内のPlayListMark()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図42は、PlayListMarkのシンタクスを示す図である。図42に示したPlayListMarkのシンタクスについて説明するに、version_numberは、このPlayListMark()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0193】lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト

数を示す32ビットの符号なし整数である。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークの個数を示す16ビットの符号なし整数である。number_of_PlayList_marks は、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図43に示すテーブルに従って符号化される。

【0194】mark_time_stampの32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図44に示すように、PlayList()において定義されるCPI_typeによって異なる。PlayItem_idは、マークが置かれているところのPlayItemを指定する8ビットのフィールドである。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される(図25参照)。

【0195】character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示した値に対応する。name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。Mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どのような値が設定されても良い。

【0196】ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される(後述)。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのマークにはサムネイル画像が付加されていない事を示す。

【0197】次に、Clip information fileについて説明する。zzzzz.clpi (Clip information fileファイル)は、図45に示すように6個のオブジェクトから構成される。それらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()、およびMakerPrivateData()である。AVストリーム(Clip AVストリームまたはBridge-Clip AV stream)とそれに対応するClip Information ファイルは、同じ数字列の"zzzzz"が使用される。

【0198】図45に示したzzzzz.clpi (Clip information fileファイル)のシンタクスについて説明するに、ClipInfo_Start_addressは、zzzzz.clpi ファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0199】STC_Info_Start_addressは、zzzzz.clpi ファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。ProgramInfo_Start_addressは、zzzzz.clpi ファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。CPI_Start_addressは、zzzzz.clpi ファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0200】ClipMark_Start_addressは、zzzzz.clpi ファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。MakerPrivateData_Start_addressは、zzzzz.clpi ファイルの先頭のバイトからの相対バイト数を単位として、MakerPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word (パディングワード) は、zzzzz.clpi ファイルのシンタックスにしたがって挿入される。N1, N2, N3, N4, およびN5は、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしても良い。

【0201】次に、ClipInfoについて説明する。図46は、ClipInfoのシンタックスを示す図である。ClipInfo()は、それに対応するAVストリームファイル (Clip AVストリームまたはBridge-Clip AVストリームファイル) の属性情報をストアする。

【0202】図46に示したClipInfoのシンタックスについて説明するに、version_numberは、このClipInfo()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す32ビットの符号なし整数である。Clip_stream_typeの8ビットのフィールドは、図47に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

【0203】offset_SPNの32ビットのフィールドは、AVストリーム (Clip AVストリームまたはBridge-Clip AVストリーム) ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録される時、このoffset_SPNは0でなければならない。

【0204】図48に示すように、AVストリームファイルのはじめの部分が編集によって消去された時、offset_SPNは、ゼロ以外の値をとっても良い。本実施の形態では、offset_SPNを参照する相対ソースパケット番号 (相対アドレス) が、しばしば、RSPN_xxx (xxxは変形する。例、RSPN_EP_start) の形式でシンタックスの中に記

述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

【0205】AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数 (SPN_xxx) は、次式で算出される。

$$SPN_xxx = RSPN_xxx - offset_SPN$$

図48に、offset_SPNが4である場合の例を示す。

【0206】TS_recording_rateは、24ビットの符号なし整数であり、この値は、DVRドライブ (書き込み部22) へまたはDVRドライブ (読み出し部28) からのAVストリームの必要な入出力のビットレートを与える。record_time_and_dateは、Clipに対応するAVストリームが記録された時の日時をストアする56ビットのフィールドであり、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

【0207】durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

【0208】time_controlled_flag:のフラグは、AVストリームファイルの記録モードを示す。このtime_controlled_flagが1である場合、記録モードは、記録してから時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$TS_average_rate * 192 / 188 * (t - start_time) - \alpha \leq size_clip(t)$$

$$\leq TS_average_rate * 192 / 188 * (t - start_time) + \alpha$$

ここで、TS_average_rateは、AVストリームファイルのトランスポートストリームの平均ビットレートをbytes/second の単位で表したものである。

【0209】また、上式において、tは、秒単位で表される時間を示し、start_timeは、AVストリームファイルの最初のソースパケットが記録された時の時刻であり、秒単位で表される。size_clip(t)は、時刻tにおけるAVストリームファイルのサイズをバイト単位で表したものであり、例えば、start_timeから時刻tまでに10個のソースパケットが記録された場合、size_clip(t)は10 * 192バイトである。αは、TS_average_rateに依存する定数である。

【0210】time_controlled_flagが0にセットされている場合、記録モードは、記録の時間経過とAVストリームのファイルサイズが比例するように制御していないこ

とを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

【0211】TS_average_rateは、time_controlled_flagが1にセットされている場合、この24ビットのフィールドは、上式で用いているTS_average_rateの値を示す。time_controlled_flagが0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。まずトランスポートレートをTS_recording_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しない事によって、間欠的にトランスポートパケットを符号化する。

【0212】RSPN_arrival_time_discontinuityの32ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットからClipInfo() において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した

$$SPN_{xxx} = RSPN_{xxx} - offset_SPN$$

に基づいて算出される。

【0213】reserved_for_system_useの144ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが1である時、format_identifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが1である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが1である場合、transport_stream_IDのフィールドが有効であることを示す。is_service_ID_validのフラグが1である場合、service_IDのフィールドが有効であることを示す。

【0214】is_country_code_validのフラグが1である時、country_codeのフィールドが有効であることを示す。format_identifierの32ビットフィールドは、トランスポートストリームの中でregistration_descriptor (ISO/IEC13818-1で定義されている) が持つformat_identifierの値を示す。original_network_IDの16ビットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの16ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

【0215】service_IDの16ビットフィールドは、トランスポートストリームの中で定義されているservice_IDの値を示す。country_codeの24ビットのフィールド

は、ISO3166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO8859-1で符号化される。例えば、日本は"JPN"と表され、"0x4A 0x500 x4E"と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値'0xFF'がセットされる。

【0216】format_identifier、original_network_ID、transport_stream_ID、service_ID、country_code、およびstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI (サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコードシステムの初期設定を行うために用いることが可能である。

【0217】次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点 (システムタイムベースの不連続点) を含まない時間区間をSTC_sequenceと称し、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図50は、連続なSTC区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない (ただし、後述するように、Clipの最大時間長は制限されている)。従って、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC_sequenceに分割される。

【0218】STC_Infoは、STCの不連続 (システムタイムベースの不連続) が発生する場所のアドレスをストアする。図51を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除くk番目 (k>=0) のSTC_sequenceは、k番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、(k+1)番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

【0219】図52は、STC_Infoのシンタクスを示す図である。図52に示したSTC_Infoのシンタクスについて説明するに、version_numberは、このSTC_Info()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0220】lengthは、このlengthフィールドの直後か

らSTC_Info()の最後までSTC_Info()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドはゼロにセットしても良い。CPI()のCPI_typeがEP_map typeを示す場合、num_of_STC_sequencesは1以上の値でなければならない。

【0221】num_of_STC_sequencesの8ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-loopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、0から開始される。

【0222】RSPN_STC_startの32ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしても良い。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述した $SPN_xxx = RSPN_xxx - offset_SPN$ により算出される。

【0223】次に、図45に示したzzzzz.clipのシンタクス内のProgramInfoについて説明する。図53を参照しながら説明するに、ここでは、Clipの中で次の特徴をもつ時間区間をprogram_sequenceと呼ぶ。まず、PCR_PIDの値が変わらない。次に、ビデオエレメンタリストリームの数が変化しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。さらに、オーディオエレメンタリストリームの数が変化しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

【0224】program_sequenceは、同一の時刻において、ただ1つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ1つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

【0225】図54は、ProgramInfoのシンタクスを示す図である。図54に示したProgramInfoのシンタクスを説明するに、version_numberは、このProgramInfo()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符

号化されなければならない。

【0226】lengthは、このlengthフィールドの直後からProgramInfo()の最後までProgramInfo()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドはゼロにセットされても良い。CPI()のCPI_typeがEP_map typeを示す場合、number_of_programsは1以上の値でなければならない。

【0227】number_of_program_sequencesの8ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが変化しない場合、number_of_program_sequencesは1をセットされなければならない。RSPN_program_sequence_startの32ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

【0228】RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$SPN_xxx = RSPN_xxx - offset_SPN$$

により算出される。シンタクスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

【0229】PCR_PIDの16ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの8ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number_of_audiosの8ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【0230】audio_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【0231】なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番

は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

【0232】図54は、図54に示したProgramInfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図55に示したVideoCodingInfoのシンタクスを説明するに、video_formatの8ビットフィールドは、図56に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

【0233】frame_rateの8ビットフィールドは、図57に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの8ビットフィールドは、図58に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

【0234】図59は、図54に示したProgramInfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図59に示したAudioCodingInfoのシンタクスを説明するに、audio_codingの8ビットフィールドは、図60に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

【0235】audio_component_typeの8ビットフィールドは、図61に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

【0236】次に、図45に示したzzzzz.clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには2つのタイプがあり、それらはEP_mapとTU_mapである。図63に示すように、CPI()の中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図64に示すように、CPI()の中のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。1つのAVストリームは、1つのEP_mapまたは1つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならない。

【0237】図65は、CPIのシンタクスを示す図である。図65に示したCPIのシンタクスを説明するに、version_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までのCPI()のバイト数を示す32ビットの符号なし整数である。CPI_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタイプを表す。

【0238】次に、図65に示したCPIのシンタクス内

のEP_mapについて説明する。EP_mapには、2つのタイプがあり、それはビデオストリーム用のEP_mapとオーディオストリーム用のEP_mapである。EP_mapの中のEP_map_typeが、EP_mapのタイプを区別する。Clipが1つ以上のビデオストリームを含む場合、ビデオストリーム用のEP_mapが使用されなければならない。Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP_mapが使用されなければならない。

【0239】ビデオストリーム用のEP_mapについて図67を参照して説明する。ビデオストリーム用のEP_mapは、stream_PID、PTS_EP_start、およびRSPN_EP_startというデータを持つ。stream_PIDは、ビデオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、ビデオストリームのシーケンスヘッダから始めるアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットの第1バイト目を含むソースポケットのアドレスを示す。

【0240】EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【0241】オーディオストリーム用のEP_mapは、stream_PID、PTS_EP_start、およびRSPN_EP_startというデータを持つ。stream_PIDは、オーディオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startで参照されるアクセスユニットの第1バイト目を含むソースポケットのアドレスを示す。

【0242】EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【0243】EP_mapとSTC_Infoの関係を説明するに、1つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較する事により、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる(図68を参照)。EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP_map_for_one_stream_PID()を持たなければならない。図69に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_f

or_one_stream_PIDを持たねばならない。

【0244】図70は、EP_mapのシンタックスを示す図である。図70に示したEP_mapのシンタックスを説明するに、EP_typeは、4ビットのフィールドであり、図71に示すように、EP_mapのエントリーポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP_typeは0('video')にセットされなければならない。または、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP_typeは1('audio')にセットされなければならない。

【0245】number_of_stream_PIDsの16ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数にもつfor-loopのループ回数を示す。stream_PID(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリストリーム（ビデオまたはオーディオストリーム）を伝送するトランスポートパケットのPIDを示す。EP_typeが0('video')に等しい場合、そのエレメンタリストリームはビデオストリームでなければならない。また、EP_typeが1('audio')に等しい場合、そのエレメンタリストリームはオーディオストリームでなければならない。

【0246】num_EP_entries(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_Start_address(k)：この32ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第1バイト目からの大きさで示される。

【0247】padding_wordは、EP_map()のシンタックスにしたがって挿入されなければならない。XとYは、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取っても良い。

【0248】図72は、EP_map_for_one_stream_PIDのシンタックスを示す図である。図72に示したEP_map_for_one_stream_PIDのシンタックスを説明するに、PTS_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。EP_typeが1('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。

【0249】RSPN_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0('video')に等しい場

合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースポケットの相対アドレスを示す。または、EP_typeが1('audio')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第1バイト目を含むソースポケットの相対アドレスを示す。

【0250】RSPN_EP_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、
$$SPN_{xxx} = RSPN_{xxx} - offset_SPN$$

により算出される。シンタックスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

【0251】次に、TU_mapについて、図73を参照して説明する。TU_mapは、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、offset_timeから一定の単位に分割される。その単位を、time_unitと称する。

【0252】AVストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startと称する。TU_map_time_axis上において、k (k>=0)番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

$$TU_start_time(k) = offset_time + k * time_unit_size$$

TU_start_time(k)は、45kHzの精度を持つ。

【0253】図75は、TU_mapのシンタックスを示す図である。図75に示したTU_mapのシンタックスを説明するに、offset_timeの32bit長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset_timeはゼロにセットされなければならない。

【0254】time_unit_sizeの32ビットフィールドは、time_unitの大きさを与えるものであり、それは27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。time_unit_sizeは、1秒以下 (time_unit_size<=45000) にすることが良い。number_of_time_unit_entriesの32ビットフィールドは、TU_map()の中にストアされているtime_u

nitのエントリー数を示す。

【0255】RSPN_time_unit_startの32ビットフィールドは、AVストリームの中でそれぞれのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタックスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k+1)番目のtime_unitの中にソースパケットが何も無い場合、(k+1)番目のRSPN_time_unit_startは、k番目のRSPN_time_unit_startと等しくなければならない。

【0256】図45に示したzzzzz.clipのシンタックス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置1）によってセットされるものであり、ユーザによってセットされるものではない。

【0257】図75は、ClipMarkのシンタックスを示す図である。図75に示したClipMarkのシンタックスを説明するに、version_numberは、このClipMark()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0258】lengthは、このlengthフィールドの直後からClipMark()の最後までClipMark()のバイト数を示す32ビットの符号なし整数である。number_of_Clip_marksは、ClipMarkの中にストアされているマークの個数を示す16ビットの符号なし整数。number_of_Clip_marksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図76に示すテーブルに従って符号化される。

【0259】mark_time_stampは、32ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図77に示すように、Playlist()の中のCPI_typeにより異なる。

【0260】STC_sequence_idは、CPI()の中のCPI_typeがEP_map_typeを示す場合、この8ビットのフィールドは、マークが置かれているところのSTC連続区間のSTC_sequence_idを示す。CPI()の中のCPI_typeがTU_map_typeを示す場合、この8ビットのフィールドは何も意味を持たず、ゼロにセットされる。character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

【0261】name_lengthの8ビットフィールドは、Mar

k_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0262】ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

【0263】MakersPrivateDataについては、図22を参照して既に説明したので、その説明は省略する。

【0264】次に、サムネイルインフォメーション(Thumbnail Information)について説明する。サムネイル画像は、menu.thmbファイルまたはmark.thmbファイルにストアされる。これらのファイルは同じシンタックス構造であり、ただ1つのThumbnail()を持つ。menu.thmbファイルは、メニューサムネイル画像、すなわちVolumeを代表する画像、および、それぞれのPlaylistを代表する画像をストアする。すべてのメニューサムネイルは、ただ1つのmenu.thmbファイルにストアされる。

【0265】mark.thmbファイルは、マークサムネイル画像、すなわちマーク点を表すピクチャをストアする。すべてのPlaylistおよびClipに対するすべてのマークサムネイルは、ただ1つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は1つのtn_blockに格納される。1つの画像データは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。1つのサムネイル画像のバイト長は可変である。

【0266】図78は、menu.thmbとmark.thmbのシンタックスを示す図であり、図79は、図78に示したmenu.thmbとmark.thmbのシンタックス内のThumbnailのシンタックスを示す図である。図79に示したThumbnailのシンタックスについて説明するに、version_numberは、このThumbnail()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0267】lengthは、このlengthフィールドの直後からThumbnail()の最後までMakersPrivateData()のバイト数を示す32ビットの符号なし整数である。tn_block

s_start_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。相対バイト数はゼロからカウントされる。number_of_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリ数を与える16ビットの符号なし整数である。

【0268】tn_block_sizeは、1024バイトを単位として、1つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size=1ならば、それは1つのtn_blockの大きさが1024バイトであることを示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエントリ数を表す16ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデックス番号を表す16ビットの符号なし整数である。thumbnail_indexとして、0xFFFFという値を使用してはならない。thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlaylist()、PlaylistMark()、およびClipMark()の中のref_thumbnail_indexによって参照される。

【0269】thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図80に示すような値をとる。表中のDCFとPNGは"menu.thmb"内でのみ許される。マークサムネイルは、値"0x00" (MPEG-2 Video I-picture)をとらなければならない。

【0270】picture_data_sizeは、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_blockのtn_block番号を表す16ビットの符号なし整数である。サムネイル画像データの先頭は、tn_blockの先頭と一致していなければならない。tn_block番号は、0から始まり、tn_blockのfor-ループ中の変数kの値に関係する。

【0271】x_picture_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。y_picture_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。tn_blockは、サムネイル画像がストアされる領域である。Thumbnail()の中のすべてのtn_blockは、同じサイズ(固定長)であり、その大きさはtn_block_sizeによって定義される。

【0272】図81は、サムネイル画像データがどのようにtn_blockに格納されるかを模式的に表した図である。図81のように、各サムネイル画像データはtn_blockの先頭から始まり、1 tn_blockを超える大きさの場合は、連続する次のtn_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、

削除といった編集に対して簡便な処理により対応する事ができるようになる。

【0273】次に、AVストリームファイルについて説明する。AVストリームファイルは、"M2TS"ディレクトリ(図14)にストアされる。AVストリームファイルには、2つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

【0274】まず、DVR MPEG-2 トランスポートストリームについて説明する。DVR MPEG-2トランスポートストリームの構造は、図82に示すようになっている。AVストリームファイルは、DVR MPEG2トランスポートストリームの構造を持つ。DVR MPEG2トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144 バイト (2048*3 バイト)である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、192バイト長である。1つのソースパケットは、TP_extra_headerとトランスポートパケットから成る。TP_extra_headerは、4バイト長であり、またトランスポートパケットは、188バイト長である。

【0275】1つのAligned unitは、32個のソースパケットから成る。DVR MPEG2トランスポートストリームの中の最後のAligned unitも、また32個のソースパケットから成る。よって、DVR MPEG2トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が32の倍数でない時、ヌルパケット(PID=0x1FFFのトランスポートパケット)を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG2トランスポートストリームに余分な情報を付加してはならない。

【0276】図83に、DVR MPEG-2トランスポートストリームのレコーダモデルを示す。図83に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

【0277】MPEG-2トランスポートストリームの入力タイミングについて説明する。入力MPEG2トランスポートストリームは、フルトランスポートストリームまたはパーシャルトランスポートストリームである。入力されるMPEG2トランスポートストリームは、ISO/IEC13818-1またはISO/IEC13818-9に従っていないなければならない。MPEG2トランスポートストリームのi番目のバイトは、T-STD (ISO/IEC 13818-1で規定されるTransport stream system target decoder)とソースパケットタイザへ、時刻*t*(i)に同時に入力される。Rpkは、トランスポートパケットの入力レートの瞬時的な最大値である。

【0278】27MHz PLL52は、27MHzクロックの周波数を発生する。27MHzクロックの周波数は、MPEG-2トランスポートストリームのPCR (Program Clock Reference)の値にロックされる。arrival_time_clock_counter53は、27MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival_time_clock(i)は、時刻t(i)におけるArrival time clock counterのカウント値である。

【0279】source_packetizer54は、すべてのトランスポートパケットにTP_extra_headerを付加し、ソースパケットを作る。Arrival_time_stampは、トランスポートパケットの第1バイト目がT-STDとソースパケットアイザの両方へ到着する時刻を表す。Arrival_time_stamp(k)は、次式で示されるようにArrival_time_clock(k)のサンプル値であり、ここで、kはトランスポートパケットの第1バイト目を示す。

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 2^{30}$$

【0280】2つの連続して入力されるトランスポートパケットの時間間隔が、 $2^{30}/27000000$ 秒(約40秒)以上になる場合、その2つのトランスポートパケットのarrival_time_stampの差分は、 $2^{30}/27000000$ 秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

【0281】smoothing_buffer55は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファは、オーバーフローしてはならない。Rmaxは、スムージングバッファが空でない時のスムージングバッファからのソースパケットの出力ビットレートである。スムージングバッファが空である時、スムージングバッファからの出力ビットレートはゼロである。

【0282】次に、DVR MPEG-2トランスポートストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

$$R_{\max} = \text{TS_recording_rate} \cdot 192/188$$

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

【0283】入力トランスポートストリームがSESFトランスポートストリームの場合、Rpkは、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateに等しくなければならない。入力トランスポートストリームがSESFトランスポートストリームでない場合、この値はMPEG-2 transport streamのデスクリプター、例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptorなど、において定義される値を参照しても良い。

【0284】smoothing_buffer_sizeは、入力トランス

ポートストリームがSESFトランスポートストリームの場合、スムージングバッファの大きさはゼロである。入力トランスポートストリームがSESFトランスポートストリームでない場合、スムージングバッファの大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptorなどにおいて定義される値を参照しても良い。

【0285】記録機(レコーダ)および再生機(プレーヤ)は、十分なサイズのバッファを用意しなければならない。デフォルトのバッファサイズは、1536 bytesである。

【0286】次に、DVR MPEG-2トランスポートストリームのプレーヤモデルについて説明する。図84は、DVR MPEG-2トランスポートストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

【0287】27MHz X-tal61は、27MHzの周波数を発生する。27MHz周波数の誤差範囲は、 ± 30 ppm (27000000 ± 810 Hz)でなければならない。arrival_time_clock_counter62は、27MHzの周波数のパルスをカウントするバイナリーカウンターである。Arrival_time_clock(i)は、時刻t(i)におけるArrival time clock counterのカウント値である。

【0288】smoothing_buffer64において、Rmaxは、スムージングバッファがフルでない時のスムージングバッファへのソースパケットの入力ビットレートである。スムージングバッファがフルである時、スムージングバッファへの入力ビットレートはゼロである。

【0289】MPEG-2トランスポートストリームの出力タイミングを説明するに、現在のソースパケットのarrival_time_stampがarrival_time_clock(i)のLSB 30ビットの値と等しい時、そのソースパケットのトランスポートパケットは、スムージングバッファから引き抜かれる。Rpkは、トランスポートパケットレートの瞬時的な最大値である。スムージングバッファは、アンダーフローしてはならない。

【0290】DVR MPEG-2トランスポートストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランスポートストリームのレコーダモデルのパラメータと同一である。

【0291】図85は、Source packetのシンタクスを示す図である。transport_packet()は、ISO/IEC 13818-1で規定されるMPEG-2トランスポートパケットである。図85に示したSource packetのシンタクス内のTP_Extra_headerのシンタクスを図86に示す。図86に示したTP_Extra_headerのシンタクスについて説明するに、copy_permission_indicatorは、トランスポートパケットのペイロードのコピー制限を表す整数である。コピー制限

は、copy free、no more copy、copy once、またはcopy prohibitedとすることができる。図87は、copy_permission_indicatorの値と、それらによって指定されるモードの関係を示す。

【0292】copy_permission_indicatorは、すべてのトランスポートパケットに付加される。IEEE1394デジタルインターフェースを使用して入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、IEEE1394 isochronous packet headerの中のEMI (Encryption Mode Indicator)の値に関連付けても良い。IEEE1394デジタルインターフェースを使用しないで入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、トランスポートパケットの中に埋め込まれたCCIの値に関連付けても良い。アナログ信号入力をセルフエンコードする場合、copy_permission_indicatorの値は、アナログ信号のCGMS-Aの値に関連付けても良い。

【0293】arrival_time_stampは、次式

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 2^{30}$$

において、arrival_time_stampによって指定される値を持つ整数値である。

【0294】Clip AVストリームの定義をするに、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。arrival_time_clock(i)は、Clip AVストリームの中で連続して増加しななければならない。Clip AVストリームの中にシステムタイムベース (STCベース) の不連続点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しななければならない。

【0295】Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、26時間 でなければならない。この制限は、MPEG2トランスポートストリームの中にシステムタイムベース (STCベース) の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS(Presentation Time Stamp)が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間)と規定している。

【0296】Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければならない。かつ後述するDVR-STDに従わなければならない。

【0297】本実施の形態においては、編集におけるPl

aylItem間のビデオとオーディオのシームレス接続をサポートする。PlaylItem間をシームレス接続にすることは、プレーヤ/レコーダに"データの連続供給"と"シームレスな復号処理"を保証する。"データの連続供給"とは、ファイルシステムが、デコーダにバッファのアンダーフローを起こさせる事のないように必要なビットレートでデータを供給する事を保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるように、データが十分な大きさの連続したブロック単位でストアされるようにする。

【0298】"シームレスな復号処理"とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせる事なく、ディスクに記録されたオーディオビデオデータを表示できることである。

【0299】シームレス接続されているPlaylItemが参照するAVストリームについて説明する。先行するPlaylItemと現在のPlaylItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlaylItemにおいて定義されているconnection_conditionフィールドから判断することができる。PlaylItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

【0300】図88は、Bridge-Clipを使用する場合の先行するPlaylItemと現在のPlaylItemの関係を示している。図88においては、プレーヤが読み出すストリームデータが、影をつけて示されている。図88に示したTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータから成る。

【0301】TS1のClip1の影を付けられたストリームデータは、先行するPlaylItemのIN_time (図88においてIN_time1で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

【0302】また、図88におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータから成る。TS2に含まれるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照され

るソースパケットから、現在のPlayItemのOUT_time (図88においてOUT_time2で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【0303】図89は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影をつけて示されている。図89におけるTS1は、Clip1 (Clip AVストリーム)の影を付けられたストリームデータから成る。TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図89においてIN_time1で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。また、図89におけるTS2は、Clip2 (Clip AVストリーム)の影を付けられたストリームデータから成る。

【0304】TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT_time (図89においてOUT_time2で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【0305】図88と図89において、TS1とTS2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、まず、TS1とTS2の中に含まれるプログラムの数は、1でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、1でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1および/またはTS2の中に、上記以外のエレメンタリストリームまたはプライベートストリームが含まれていても良い。

【0306】ビデオビットストリームの制限について説明する。図90は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT_time1 (Clip1のOUT_time) の後とIN_time2 (Clip2のIN_time) の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

【0307】図90に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図91に示す。RSPN_arrival_time_discontinuityより前のBridge-Clipのビデオストリームは、図90のClip1のOUT_time1に対応するピクチャまでの符号化ビデオス

トリームから成る。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。

【0308】同様に、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図90のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成る。そして、そのビデオストリームは、正しくデコード開始する事ができて、これに続くClip2のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならない、それ以外のピクチャはオリジナルのClipからコピーすることができる。

【0309】図90に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図92に示す。Clip1のビデオストリームは、図90のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成り、それは、1つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。同様に、Clip2のビデオストリームは、図90のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成り、それは、1つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。

【0310】ビデオストリームの符号化制限について説明するに、まず、TS1とTS2のビデオストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてI-ピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

【0311】ビットストリームの中で定義されるビデオプレゼンテーションユニット (フレームまたはフィールド) は、接続点を挟んで連続でなければならない。接続点において、フレームまたはフィールドのギャップがあってはならない。接続点において、トップ?ボトムのフィールドシーケンスは連続でなければならない。3-2フィールドダウンを使用するエンコードの場合は、"top_field_first" および "repeat_first_field" フラグを書き換える必要があるかもしれない、またはフィールドギャップの発生を防ぐために局所的に再エンコードするようにしても良い。

【0312】オーディオビットストリームの符号化制限について説明するに、TS1とTS2のオーディオのサンプリング周波数は、同じでなければならない。TS1とTS2のオーディオの符号化方法 (例、MPEG1レイヤ2、AC-3、SMP/LPCM、AAC) は、同じでなければならない。

【0313】次に、MPEG-2トランスポートストリームの符号化制限について説明するに、TS1のオーディオストリームの最後のオーディオフレームは、TS1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2のオーディオストリームの最初のオーディオフレームは、TS2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

【0314】接続点において、オーディオプレゼンテーションユニットのシーケンスにギャップがあってはならない。図93に示すように、2オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあっても良い。TS2のエレメンタリーストリームを送送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

【0315】ClipおよびBridge-Clipの制限について説明するに、TS1とTS2は、それぞれの中にアライバルタイムベースの不連続点を含んではいならない。

【0316】以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ1つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

【0317】BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。

【0318】PlayItemの制限について説明するに、先行するPlayItemのOUT_time (図88、図89において示されるOUT_time1) は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time (図88、図89において示されるIN_time2) は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

【0319】Bridge-Clipを使用する場合のデータアロケーションの制限について、図94を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) とClip2 (Clip AVストリームファイル) に接続されるBridge

-Clip AVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

【0320】RSPN_exit_from_previous_Clip以前のClip1 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip AVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip2 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

【0321】Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図95を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) の最後の部分とClip2 (Clip AVストリームファイル) の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

【0322】Clip1 (Clip AVストリームファイル) の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。Clip2 (Clip AVストリームファイル) の最初のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。

【0323】次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG2トランスポートストリームの生成および検証の際におけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるAVストリームの生成および検証の際におけるデコード処理をモデル化するための概念モデルでもある。

【0324】DVR-STDモデルを図96に示す。図96に示したモデルには、DVR MPEG-2トランスポートストリームプレーヤモデルが構成要素として含まれている。n, T, Bn, MBn, EBn, TBSys, BSys, Rxn, Rbxn, Rxsys, Dn, Dsys, OnおよびPn(k)の表記方法は、ISO/IEC13818-1のT-S TDに定義されているものと同じである。すなわち、次の通りである。nlは、エレメンタリーストリームのインデックス番号である。TBnは、エレメンタリーストリームnのトランスポートバッファである。

【0325】MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBSysは、復号中のプログラムのシス

テム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

【0326】Rsysは、データがTBSysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシステム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

【0327】DVR-STDのデコーディングプロセスについて説明する。単一のDVR MPEG-2トランスポートストリームを再生している間は、トランスポートパケットをTB1、TBnまたはTBSysのバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB1、MB1、EB1、TBn、Bn、TBSysおよびBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

【0328】シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図88に示した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

【0329】図97は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャートを示す。所定のAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る間には、TS2のアライバルタイムベースの時間軸（図97においてATC2で示される）は、TS1のアライバルタイムベースの時間軸（図97においてATC1で示される）と同じでない。

【0330】また、TS2のシステムタイムベースの時間軸（図97においてSTC2で示される）は、TS1のシステムタイムベースの時間軸（図97においてSTC1で示される）と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあっても良い。

【0331】DVR-STD への入力タイミングについて説明する。時刻T1までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-

STDのTB1、TBn またはTBSysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

【0332】TS1の残りのパケットは、TS_recording_rate(TS1)のビットレートでDVR-STDのTBnまたはTBSysのバッファへ入力されなければならない。ここで、TS_recording_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻T2である。従って、時刻T1からT2までの区間では、ソースパケットのarrival_time_stampは無視される。

【0333】N1をTS1の最後のビデオパケットに続くTS1のトランスポートパケットのバイト数とすると、時刻T1乃至T2までの時間DT1は、N1バイトがTS_recording_rate(TS1)のビットレートで入力終了するために必要な時間であり、次式により算出される。

$$DT1 = T_2 - T_1 = N1 / TS_recording_rate$$

(TS1)時刻T1乃至T2までの間は、RXnとRsysの値は共に、TS_recording_rate(TS1)の値に変化する。このルール以外のバッファリング動作は、T-STDと同じである。

【0334】T2の時刻において、arrival time clock counterは、TS2の最初のソースパケットのarrival_time_stampの値にリセットされる。DVR-STDのTB1、TBn またはTBSysのバッファへの入力タイミングは、TS2のソースパケットのarrival_time_stampによって決定される。RXnとRsysは共に、T-STDにおいて定義されている値に変化する。

【0335】付加的なオーディオバッファリングおよびシステムデータバッファリングについて説明するに、オーディオデコーダとシステムデコーダは、時刻T1からT2までの区間の入力データを処理することができるように、T-STDで定義されるバッファ量に加えて付加的なバッファ量（約1秒分のデータ量）が必要である。

【0336】ビデオのプレゼンテーションタイミングについて説明するに、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC1は、TS1のシステムタイムベースの時間軸（図97ではSTC1と図示されている）とし、STC2は、TS2のシステムタイムベースの時間軸（図97ではSTC2と図示されている。正確には、STC2は、TS2の最初のPCRがT-STDに入力した時刻から開始する。）とする。

【0337】STC1とSTC2の間のオフセットは、次のように決定される。PTS_{1_end}は、TS1の最後のビデオプレゼンテーションユニットに対応するSTC1上のPTSであり、PTS_{2_start}は、TS2の最初のビデオプレゼンテーションユニットに対応するSTC2上のPTSであり、T_{pp}は、TS1の最後のビデオプレゼンテーションユニットの表示期間とすると、2つのシステムタイムベースの間のオフセットSTC_deltaは、次式により算出される。

$$STC_delta = PTS_{end}^{1} + T_{pp} - PTS_{start}^{2}$$

【0338】オーディオのプレゼンテーションのタイミングについて説明するに、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあっても良く、それは0乃至2オーディオフレーム未満である（図97に図示されている“audio overlap”を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに同期することは、プレーヤ側により設定されることである。

【0339】DVR-STDのシステムタイムクロックについて説明するに、時刻 T_5 において、TS1の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻 T_2 から T_5 の間にオーバーラップしていても良い。この区間では、DVR-STDは、システムタイムクロックを古いタイムベースの値（STC1）と新しいタイムベースの値（STC2）の間で切り替える。STC2の値は、次式により算出される。

$$STC2 = STC1 - STC_delta$$

【0340】バッファリングの連続性について説明する。STC1¹_{video_end}は、TS1の最後のビデオパケットの最後のバイトがDVR-STDのTB1へ到着する時のシステムタイムベースSTC1上のSTCの値である。STC2²_{video_start}は、TS2の最初のビデオパケットの最初のバイトがDVR-STDのTB1へ到着する時のシステムタイムベースSTC2上のSTCの値である。STC2¹_{video_end}は、STC1¹_{video_end}の値をシステムタイムベースSTC2上の値に換算した値である。STC2¹_{video_end}は、次式により算出される。

$$STC2^1_{video_end} = STC1^1_{video_end} - STC_delta$$

【0341】DVR-STDに従うために、次の2つの条件を満たす事が要求される。まず、TS2の最初のビデオパケットのTB1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$STC2^2_{video_start} > STC2^1_{video_end} + \Delta T1$$

この不等式が満たされるように、Clip1および、または、Clip2の部分的なストリームを再エンコードおよび、または、再多重化する必要がある場合は、その必要に応じて行われる。

【0342】次に、STC1とSTC2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS1からのビデオパケットの入力とそれに続くTS2からのビデオパケットの入力は、ビデオバッファをオーバーフローおよびアンダーフローさせてはならない。

【0343】このようなシンタクス、データ構造、規則に基づく事により、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生

できるようにすることができる。

【0344】なお、本実施の形態は、多重化ストリームとしてMPEG2トランスポートストリームを例にして説明しているが、これに限らず、MPEG2プログラムストリームや米国のDirecTVサービス（商標）で使用されているDSSトランスポートストリームについても適用することが可能である。

【0345】次に、図98は、PlayListファイルの別の例を示す。図98と図23のシンタクスの大きな違いは、UIAppInfoPlayList()をストアしている場所である。図98の例では、UIAppInfoPlayList()がPlayList()の中から外に出されているので、UIAppInfoPlayList()の将来の情報拡張が比較的容易に行えるようになる。

【0346】version_numberは、このサムネールヘッダ情報ファイルのバージョンナンバーを示す4個の数字である。

【0347】PlayList_start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayList()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0348】PlayListMark_start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0349】MakersPrivateData_start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0350】図99は、図98のPlayListファイルの中のUIAppInfoPlayListのシンタクスを示す。PlayList_service_typeは、PlayListファイルのタイプを示す。その一例は、図26に示されている。また、PlayList_service_typeは、デジタルTV放送のプログラムが示すサービスタイプと同じ意味を持たせても良い。例えば、日本のデジタルBS放送の場合、サービスタイプは、テレビサービス、音声サービス、およびデータ放送サービスの3種類を持つ。PlayListが使用するClip AVストリームが含むプログラムのサービスタイプを代表する値をPlayList_service_typeにセットする。

【0351】PlayList_character_setは、channel_name、PlayList_nameおよびPlayList_detailフィールドに符号化されているキャラクター文字の符号化方法を示す。また、これはPlayListMarkの中のmark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。

【0352】channel_numberは、そのPlayListが記録される時、ユーザによって選択された放送チャンネル番号またはサービス番号を示す。複数のPlayListが1つのPlayListにコンパインされた場合は、このフィールドはそのPlayListの代表値を示す。このフィールドが0xFFFFに

セットされている場合、このフィールドは何も意味を持たない。

【0353】channel_name_lengthは、channel_nameフィールドの中に示されるチャンネル名のバイト長を示す。このフィールドは、20以下の値である。

【0354】channel_nameは、そのPlaylistが記録される時、ユーザによって選択された放送チャンネルまたはサービスの名前を示す。このフィールドの中の左からchannel_name_lengthによって示される数のバイト数が有効なキャラクター文字であり、前記名前を示す。このフィールドの中で、それら有効なキャラクター文字に続く残りのバイトは、どんな値がセットされていても良い。複数のPlaylistが1つのPlaylistにコンバインされた場合は、このフィールドはそのPlaylistを代表する名前を示す。

【0355】Playlist_name_lengthは、Playlist_nameフィールドの中に示されるPlaylist名のバイト長を示す。

【0356】Playlist_nameは、Playlistの名前を示す。このフィールドの中の左からPlaylist_name_lengthによって示される数のバイト数が有効なキャラクター文字であり、前記名前を示す。このフィールドの中で、それら有効なキャラクター文字に続く残りのバイトは、どんな値がセットされていても良い。

【0357】Playlist_detail_lengthは、Playlist_detailフィールドの中に示されるPlaylistの詳細情報のバイト長を示す。このフィールドは、1200以下の値である。

【0358】Playlist_detailは、Playlistの詳細情報を説明するテキストを示す。このフィールドの中の左からPlaylist_detail_lengthによって示される数のバイト数が有効なキャラクター文字であり、前記テキストを示す。このフィールドの中で、それら有効なキャラクター文字に続く残りのバイトは、どんな値がセットされていても良い。

【0359】これ以外のシンタクスフィールドの意味は、図27に示す同名のフィールドと同じである。

【0360】図100は、図98のPlaylistファイルの中のPlaylist()のシンタクスを示す。図25の例と比べると、UIAppInfoPlaylist()がなくなった点が違うだけで、これ以外は基本的に同じである。

【0361】図101は、SubPlayItemのシンタクスの別例を示す。図40の例と比べると、STC_sequence_idが追加された点が大きな違いである。

【0362】STC_sequence_idは、Clip_Information_file_nameに対応するAVストリームファイル上の再生区間を特定するためのSubPath_IN_timeとSubPath_OUT_timeが参照するところのSTCのSTC_sequence_idを示す。SubPath_IN_timeとSubPath_OUT_timeは、STC_sequence_idによって指定される同じSTC連続区間上の時間を示す。

【0363】SubPlayItemにSTC_sequence_idを追加することにより、SubPlayItemが参照するAVストリームファイルがSTC不連続点を持つことが許されるようになる。

【0364】これ以外のシンタクスフィールドの意味は、図40に示す同名のフィールドと同じである。

【0365】図102は、Real Playlistの作成方法を説明するフローチャートを示す。図1の記録再生装置のブロック図を参照しながら説明する。

【0366】ステップS11で、制御部23はClip AVストリームを記録する。

【0367】ステップS12で、制御部23はClip AVストリームのEP_mapを作成可能かどうかを調べる。ステップS12で、Yesの場合はステップS13へ進み、EP_mapを作成する。ステップS12で、Noの場合はステップS14へ進み、TU_mapを作成する。

【0368】その後、ステップS15で、制御部23はPlaylistのCPI_typeをセットする。

【0369】ステップS16で、制御部23は上記Clipの全ての再生可能範囲をカバーするPlayItemからなるPlaylist()を作成する。CPI_typeがEP_mapタイプの場合は、時間情報をPTSベースでセットする、この時、Clipの中にSTC不連続点があり、Playlist()が2つ以上のPlayItemからなる場合は、PlayItem間のconnection_conditionもまた決定する。CPI_typeがTU_mapタイプの場合は、時間情報をアライバルタイムベースでセットする。

【0370】ステップS17で、制御部23はUIAppInfoPlaylist()を作成する。

【0371】ステップS18で、制御部23はPlaylistMarkを作成する。

【0372】ステップS19で、制御部23はMakersPrivateDataを作成する。

【0373】ステップS20で、制御部23はReal Playlistファイルを記録する。

【0374】このようにして、新規にClip AVストリームを記録する毎に、1つのReal Playlistファイルが作られる。

【0375】図103は、Virtual Playlistの作成方法を説明するフローチャートである。

【0376】ステップS31で、ユーザインターフェースを通して、ディスクに記録されている1つのReal Playlistが指定される。そして、そのReal Playlistの再生範囲の中から、ユーザインターフェースを通して、IN点とOUT点で示される再生区間が指定される。CPI_typeがEP_mapタイプの場合は、再生区間をPTSベースでセットし、CPI_typeがTU_mapタイプの場合は、再生区間をアライバルタイムベースでセットする。

【0377】ステップS32で、制御部23はユーザによる再生範囲の指定操作がすべて終了したか調べる。ユーザが上記指示した再生区間に続けて再生する区間を選ぶ場合はステップS31へ戻る。ステップS32でユー

ザによる再生範囲の指定操作がすべて終了した場合は、ステップS33へ進む。

【0378】ステップS33で、連続して再生される2つの再生区間の間の接続状態(connection_condition)を、ユーザがユーザインタフェースを通して決定するか、または制御部23が決定する。

【0379】ステップS34で、CPI_typeがEP_mapタイプの場合、ユーザインタフェースを通して、ユーザがサブパス(アフレコ用オーディオ)情報を指定する。ユーザがサブパスを作成しない場合はこのステップはない。

【0380】ステップS35で、制御部23はユーザが指定した再生範囲情報、およびconnection_conditionに基づいて、Playlist()を作成する。

【0381】ステップS36で、制御部23はUIAppInfoPlaylist()を作成する。

【0382】ステップS37で、制御部23はPlaylistMarkを作成する。

【0383】ステップS38で、制御部23はMakersPrivateDataを作成する。

【0384】ステップS39で、制御部23はVirtualPlaylistファイルを記録する。

【0385】このようにして、ディスクに記録されているReal Playlistの再生範囲の中から、ユーザが見たい再生区間を選択してその再生区間をグループ化したものに、1つのVirtual Playlistファイルが作られる。

【0386】図104はPlaylistの再生方法を説明するフローチャートである。

【0387】ステップS51で、制御部23はInfo.dvr, Clip Information file, Playlist fileおよびサムネールファイルの情報を取得し、ディスクに記録されているPlaylistの一覧を示すGUI画面を作成し、ユーザインタフェースを通して、GUIに表示する。

【0388】ステップS52で、制御部23はそれぞれのPlaylistのUIAppInfoPlaylist()に基づいて、Playlistを説明する情報をGUI画面に提示する。

【0389】ステップS53で、ユーザインタフェースを通して、GUI画面上からユーザが1つのPlaylistの再生を指示する。

【0390】ステップS54で、制御部23は、CPI_typeがEP_mapタイプの場合、現在のPlayItemのSTC-sequential-idとIN_timeのPTSから、IN_timeより時間的に前で最も近いエン트리ポイントのあるソースパケット番号を取得する。または制御部23は、CPI_typeがTU_mapタイプの場合、現在のPlayItemのIN_timeから、IN_timeより時間的に前で最も近いタイムユニットの開始するソースパケット番号を取得する。

【0391】ステップS55で、制御部23は上記ステップで得られたソースパケット番号からAVストリームのデータを読み出し、AVデコーダ27へ供給する。

【0392】ステップS56で、現在のPlayItemの時間

的に前のPlayItemがあった場合は、制御部23は、前のPlayItemと現在のPlayItemとの表示の接続処理をconnection_conditionに従って行う。

【0393】ステップS57で、制御部23は、CPI_typeがEP_mapタイプの場合、AVデコーダ27は、IN_timeのPTSのピクチャから表示を開始するように指示する。または、制御部23は、CPI_typeがEP_mapタイプの場合、AVデコーダ27は、IN_time以後のストリームのピクチャから表示を開始するように指示する。

【0394】ステップS58で、制御部23は、AVデコーダ27にAVストリームのデコードを続けるように指示する。

【0395】ステップS59で、制御部23は、CPI_typeがEP_mapタイプの場合、現在表示の画像が、OUT_timeのPTSの画像かを調べる。または、制御部23は、CPI_typeがTU_mapタイプの場合、現在デコードしているストリームがOUT_timeを過ぎたかを調べる。

【0396】ステップS59で、Noの場合は、ステップS60へ進む。ステップS60で現在の画像を表示して、ステップS58へ戻る。Yesの場合は、ステップS61へ進む。

【0397】ステップS61で、制御部23は、現在のPlayItemがPlaylistの中で最後のPlayItemかを調べる。Noの場合はステップS54へ戻る。Yesの場合は、Playlistの再生を終了する。

【0398】図105は、PlaylistのSubパスの再生方法を説明するフローチャートである。図105のPlaylistのサブパスの再生方法は、PlaylistのCPI_typeがEP_mapの場合のみに用いられる。このフローチャートの処理は、図104のPlaylistの再生におけるステップS54以後の処理と共に、同時に行われる。また、AVデコーダ27は同時に2本のオーディオストリームのデコードが可能であることを前提とする。

【0399】ステップS71で、制御部23は、SubPlayItemの情報を取得する。

【0400】ステップS72で、制御部23は、SubPath_IN_timeよりも時間的に前で最も近いエン트리ポイントのあるソースパケット番号を取得する。

【0401】ステップS73で、制御部23は、上記エン트리ポイントのあるソースパケット番号からサブパスのAVストリームのデータを読み出し、AVデコーダ27へ供給する。

【0402】ステップS74で、制御部23は、Mainパスの再生が、sync_PlayItem_idとsync_start_PTS_of_PlayItemで示されるピクチャになったら、サブパスのオーディオを表示を開始するようにAVデコーダ27に指示する。

【0403】ステップS75でAVデコーダ27は、サブパスのAVストリームのデコードを続ける。

【0404】ステップS76で制御部23は、現在表示

するサブパスのPTSが、SubPath_OUT_timeかを調べる。Noの場合は、ステップS77へ進む。ステップS77でサブパスの表示を続けて、ステップS75へ戻る。

【0405】ステップS76で現在表示するサブパスのPTSが、SubPath_OUT_timeの場合はサブパスの表示を終了する。

【0406】図104および図105のようにして、ユーザにより再生指示された1つのPlayListファイルのメインパスおよびサブパスの再生が行なわれる。

【0407】図106は、PlayListMarkの作成方法を説明するフローチャートを示す。図1の記録再生装置のブロック図を参照しながら説明する。

【0408】ステップS91で、制御部23はInfo.dvr、Clip Information file、PlayList fileおよびThumbnail fileの情報を取得し、ディスクに記録されているPlayListの一覧を示すGUI画面を作成し、ユーザインタフェースを通して、GUIに表示する。

【0409】ステップS92で、ユーザインタフェースを通して、ユーザが1つのPlayListの再生を制御部23に指示する。

【0410】ステップS93で、制御部23は、上記指示されたPlayListの再生を開始させる（図104参照）。

【0411】ステップS94で、ユーザインタフェースを通して、ユーザがお気に入りのシーンのところにマークのセットを制御部23に指示する。

【0412】ステップS95で、制御部23は、CPI_typeがEP_mapの場合、マークのPTSとそれが属するPlayItemのPlayItem_idを取得する。または制御部23は、CPI_typeがTU_mapの場合、マーク点のアライバルタイムを取得する。

【0413】ステップS96で、制御部23はマークの情報をPlayListMark()にストアする。

【0414】ステップS97で、制御部23は、PlayListファイルを記録媒体100に記録する。

【0415】図107は、PlayListMarkを使用した頭出し再生方法を説明するフローチャートである。図1の記録再生装置のブロック図を参照しながら説明する。

【0416】ステップS111で、制御部23はInfo.dvr、Clip Information file、PlayList fileおよびThumbnail fileの情報を取得し、ディスク（記録媒体100）に記録されているPlayListの一覧を示すGUI画面を作成し、ユーザインタフェースを通して、GUIに表示する。

【0417】ステップS112で、制御部23は、ユーザインタフェースを通して、ユーザが1つのPlayListの再生を指示する。

【0418】ステップS113で、制御部23はPlayListMarkで参照されるピクチャから生成したサムネールのリストを、ユーザインタフェースを通して、GUIに表示

する。

【0419】ステップS114で、ユーザインタフェースを通して、制御部23にユーザが再生開始点のマーク点を指定する。

【0420】ステップS115で、制御部23は、CPI_typeはEP_mapタイプの場合は、マークのPTSとそれが属するPlayItem_idを取得する。または制御部23は、CPI_typeはTU_mapタイプの場合は、マークのATS(Arrival Time Stamp)を取得する。

【0421】ステップS116で、制御部23は、CPI_typeはEP_mapタイプの場合、PlayItem_idが指すPlayItemが参照するAVストリームのSTC-sequence-idを取得する。

【0422】ステップS117で、制御部23は、CPI_typeはEP_mapタイプの場合は、上記STC-sequence-idとマークのPTSに基づいて、AVストリームをデコーダへ入力する。具体的には、このSTC-sequence-idとマーク点のPTSを用いて、図104のステップS54、ステップS55と同様の処理を行う。または制御部23は、CPI_typeはTU_mapタイプの場合は、マークのATSに基づいて、AVストリームをデコーダへ入力する。具体的には、このATSを用いて図104のステップS54、ステップS55と同様の処理を行う。

【0423】ステップS118で、制御部23は、CPI_typeがEP_mapタイプの場合は、マーク点のPTSのピクチャから表示を開始させる。または制御部23は、CPI_typeがTU_mapタイプの場合は、マーク点のATS以後のピクチャから表示を開始させる。

【0424】このように、図106のようにして、ユーザがPlayListからお気に入りのシーンの開始点等を選び、それをレコード（記録再生装置1の制御部23）はPlayListMarkに管理する。また図107のようにして、ユーザがPlayListMarkにストアされているマーク点のリストから再生開始点を選択して、プレーヤはその開始点から再生を開始する。

【0425】このようなシンタックス、データ構造、規則に基づく事により、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

【0426】1ピクチャの位置を分析できる場合、EP_mapを用い、1ピクチャの位置を分析できない場合、TU_mapを用いるようにすることで、共通のアプリケーションプログラム（ソフトウェア）で、異なるフォーマットのAVストリームを、同一の記録媒体に対して記録し、再生し、管理することが可能となる。

【0427】AVストリームを、その中身（1ピクチャの位置）を分析して記録媒体に記録する場合（コグニザント記録する場合）、TU_mapを使用し、その中身（1ピ

クチャの位置)を分析せずに、そのまま記録媒体に記録する場合(ノンコグニザント記録する場合)、EP_mapを使用するなどして、共通のアプリケーションプログラムで、AVデータを、同一の記録媒体に記録し、再生し、管理することができる。

【0428】従って、例えば、スクランブルされたAVデータを、デスクランブルして(分析して)記録媒体に記録する場合、TU_mapを使用し、デスクランブルせずに(分析せずに)、そのまま記録媒体に記録する場合、EP_mapを使用するなどして、共通のアプリケーションプログラムで、AVデータを、同一の記録媒体に記録し、再生し、管理することができる。

【0429】さらに、EP_map typeとTU_map typeを、CPL_typeとして、PlayList()中に、記述できるようにしたので、Iピクチャの位置が分析できる場合、EP_mapを用い、Iピクチャの位置が分析できない場合、TU_mapを用いるようにすることができる。これにより、Iピクチャの位置を分析して記録するAVストリームデータと、分析しないで記録するAVストリームデータを、フラグを設定するだけで、共通のプログラムにより、統一して管理することが可能となる。

【0430】また、PlayListファイルやClip Informationファイルを別々に分離して記録するので、編集などによって、あるPlayListやClipの内容を変更したとき、そのファイルに関係のない他のファイルを変更する必要がない。したがって、ファイルの内容の変更が容易に行え、またその変更および記録にかかる時間を小さくできる。

【0431】さらに、最初にInfo.dvrだけを読み出して、ディスクの記録内容をユーザインタフェースへ提示し、ユーザが再生指示したPlayListファイルとそれに関連するClip Informationファイルだけをディスクから読み出すようにすれば、ユーザの待ち時間を小さくすることができる。

【0432】もし、すべてのPlayListファイルやClip Informationファイルを1つのファイルにまとめて記録すると、そのファイルサイズは非常に大きくなる。そのために、そのファイルの内容を変更して、それを記録するためにかかる時間は、個々のファイルを別々に分離して記録する場合に比べて、非常に大きくなる。本発明は、この問題を解決する。

【0433】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

【0434】この記録媒体は、図108に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク221(フロッピディスクを含む)、光ディスク222(CD-ROM (Compact Disk-Read Only Memory), DVD (Digital Versatile Disk)を含む)、光磁気ディスク223(MD (Mini-Disk)を含む)、若しくは半導体メモリ224などよりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM202や記憶部208が含まれるハードディスクなどで構成される。

【0435】なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0436】また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【0437】

【発明の効果】以上の如く、本発明の第1の情報処理装置および方法、記録媒体のプログラム、プログラム、並びに記録媒体によれば、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方を、記録方法に応じて記録するようにした。

【0438】本発明の第2の情報処理装置および方法、記録媒体のプログラム、並びにプログラムによれば、プレゼンテーションタイムスタンプと、それに対応するアクセスユニットの前記AVストリームデータ中のアドレスとの対応関係を記述する第1のテーブル、または、トランスポートパケットの到着時刻に基づいたアライバルタイムスタンプと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスとの対応関係を記述する第2のテーブルの一方が、記録方法に応じて記録されている記録媒体からそれを再生し、出力を制御するようにした。

【0439】また、本発明の第3の情報処理装置および方法、記録媒体のプログラム、プログラム、並びに第2の記録媒体によれば、主の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を記録するようにした。

【0440】本発明の第4の情報処理装置および方法、記録媒体のプログラム、並びにプログラムによれば、主

の再生パスを示す第1の情報と、前記主の再生パスと同期して再生される副の再生パスを示す第2の情報により構成される再生指定情報を記録媒体から再生し、それに基づいて出力を制御するようにした。

【0441】従って、いずれの場合においても、高速再生が可能なAVストリームと不可能なAVストリームを、共通に管理することができる。また、アフターレコーディングが可能になる。

【図面の簡単な説明】

【図1】本発明を適用した記録再生装置の一実施の形態の構成を示す図である。

【図2】記録再生装置1により記録媒体に記録されるデータのフォーマットについて説明する図である。

【図3】Real PlaylistとVirtual Playlistについて説明する図である。

【図4】Real Playlistの作成について説明する図である。

【図5】Real Playlistの削除について説明する図である。

【図6】アセンブル編集について説明する図である。

【図7】Virtual Playlistにサブパスを設ける場合について説明する図である。

【図8】Playlistの再生順序の変更について説明する図である。

【図9】Playlist上のマークとClip上のマークについて説明する図である。

【図10】メニューサムネイルについて説明する図である。

【図11】Playlistに付加されるマークについて説明する図である。

【図12】クリップに付加されるマークについて説明する図である。

【図13】Playlist、Clip、サムネイルファイルの関係について説明する図である。

【図14】ディレクトリ構造について説明する図である。

【図15】info.dvrのシンタクスを示す図である。

【図16】DVR volumeのシンタクスを示す図である。

【図17】Resume volumeのシンタクスを示す図である。

【図18】UIAppInfovolumeのシンタクスを示す図である。

【図19】Character set valueのテーブルを示す図である。

【図20】TableOfPlaylistのシンタクスを示す図である。

【図21】TableOfPlaylistの他のシンタクスを示す図である。

【図22】MakersPrivateDataのシンタクスを示す図である。

【図23】xxxxx.rplsとyyyyy.vplsのシンタクスを示す

図である。

【図24】Playlistについて説明する図である。

【図25】Playlistのシンタクスを示す図である。

【図26】Playlist_typeのテーブルを示す図である。

【図27】UIAppinfoPlaylistのシンタクスを示す図である。

【図28】図27に示したUIAppinfoPlaylistのシンタクス内のフラグについて説明する図である。

【図29】PlayItemについて説明する図である。

【図30】PlayItemについて説明する図である。

【図31】PlayItemについて説明する図である。

【図32】PlayItemのシンタクスを示す図である。

【図33】IN_timeについて説明する図である。

【図34】OUT_timeについて説明する図である。

【図35】Connection_Conditionのテーブルを示す図である。

【図36】Connection_Conditionについて説明する図である。

【図37】BridgeSequenceInfoを説明する図である。

【図38】BridgeSequenceInfoのシンタクスを示す図である。

【図39】SubPlayItemについて説明する図である。

【図40】SubPlayItemのシンタクスを示す図である。

【図41】SubPath_typeのテーブルを示す図である。

【図42】PlaylistMarkのシンタクスを示す図である。

【図43】Mark_typeのテーブルを示す図である。

【図44】Mark_time_stampを説明する図である。

【図45】zzzzz.clipのシンタクスを示す図である。

【図46】ClipInfoのシンタクスを示す図である。

【図47】Clip_stream_typeのテーブルを示す図である。

【図48】offset_SPNについて説明する図である。

【図49】offset_SPNについて説明する図である。

【図50】STC区間について説明する図である。

【図51】STC_Infoについて説明する図である。

【図52】STC_Infoのシンタクスを示す図である。

【図53】ProgramInfoを説明する図である。

【図54】ProgramInfoのシンタクスを示す図である。

【図55】VideoCondinfoのシンタクスを示す図である。

【図56】Video_formatのテーブルを示す図である。

【図57】frame_rateのテーブルを示す図である。

【図58】display_aspect_ratioのテーブルを示す図である。

【図59】AudioCondinfoのシンタクスを示す図である。

【図60】audio_codingのテーブルを示す図である。

【図61】audio_component_typeのテーブルを示す図である。

【図62】sampling_frequencyのテーブルを示す図である。

る。

- 【図63】CPIについて説明する図である。
- 【図64】CPIについて説明する図である。
- 【図65】CPIのシンタクスを示す図である。
- 【図66】CPI_typeのテーブルを示す図である。
- 【図67】ビデオEP_mapについて説明する図である。
- 【図68】EP_mapについて説明する図である。
- 【図69】EP_mapについて説明する図である。
- 【図70】EP_mapのシンタクスを示す図である。
- 【図71】EP_type valuesのテーブルを示す図である。
- 【図72】EP_map_for_one_stream_PIDのシンタクスを示す図である。
- 【図73】TU_mapについて説明する図である。
- 【図74】TU_mapのシンタクスを示す図である。
- 【図75】ClipMarkのシンタクスを示す図である。
- 【図76】mark_typeのテーブルを示す図である。
- 【図77】mark_type_stampのテーブルを示す図である。
- 【図78】menu.thmbとmark.thmbのシンタクスを示す図である。
- 【図79】Thumbnailのシンタクスを示す図である。
- 【図80】thumbnail_picture_formatのテーブルを示す図である。
- 【図81】tn_blockについて説明する図である。
- 【図82】DVR MPEG 2のトランスポートストリームの構造について説明する図である。
- 【図83】DVR MPEG 2のトランスポートストリームのレコーダモデルを示す図である。
- 【図84】DVR MPEG 2のトランスポートストリームのプレーヤモデルを示す図である。
- 【図85】source packetのシンタクスを示す図である。
- 【図86】TP_extra_headerのシンタクスを示す図である。
- 【図87】copy permission indicatorのテーブルを示す図である。
- 【図88】シームレス接続について説明する図である。
- 【図89】シームレス接続について説明する図である。
- 【図90】シームレス接続について説明する図である。
- 【図91】シームレス接続について説明する図である。

- 【図92】シームレス接続について説明する図である。
- 【図93】オーディオのオーバーラップについて説明する図である。
- 【図94】BridgeSequenceを用いたシームレス接続について説明する図である。
- 【図95】BridgeSequenceを用いないシームレス接続について説明する図である。
- 【図96】DVR STDモデルを示す図である。
- 【図97】復号、表示のタイミングチャートである。
- 【図98】PlayListファイルのシンタクスを示す図である。
- 【図99】図98のPlayListファイル中のUIApplInfoPlayListのシンタクスを示す図である。
- 【図100】図98のPlayListファイル中のPlayList()のシンタクスを示す図である。
- 【図101】SubPlayItemのシンタクスを示す図である。
- 【図102】Real PlayListの作成方法を説明するフローチャートである。
- 【図103】Virtual PlayListの作成方法を説明するフローチャートである。
- 【図104】PlayListの再生方法を説明するフローチャートである。
- 【図105】PlayListのSubパスの再生方法を説明するフローチャートである。
- 【図106】PlayListMarkの作成方法を説明するフローチャートである。
- 【図107】PlayListMarkを使用した頭出し再生方法を説明するフローチャートである。
- 【図108】媒体を説明する図である。

【符号の説明】

1 記録再生装置, 11乃至13 端子, 14 解析部, 15 AVエンコーダ, 16 マルチプレクサ, 17 スイッチ, 18 多重化ストリーム解析部, 19 ソースパケットイザ, 20 ECC符号化部, 21 変調部, 22 書き込み部, 23 制御部, 24 ユーザインタフェース, 26 デマルチプレクサ, 27 AVデコーダ, 28 読み出し部, 29復調部, 30 ECC復号部, 31 ソースパケットイザ, 32, 33 端子

【図17】

Syntax	No. of bits	Mnemonics
ResumeVolume() {		
reserved	15	bslbf
valid flag	1	bslbf
resume PlayList name	8*10	bslbf
}		

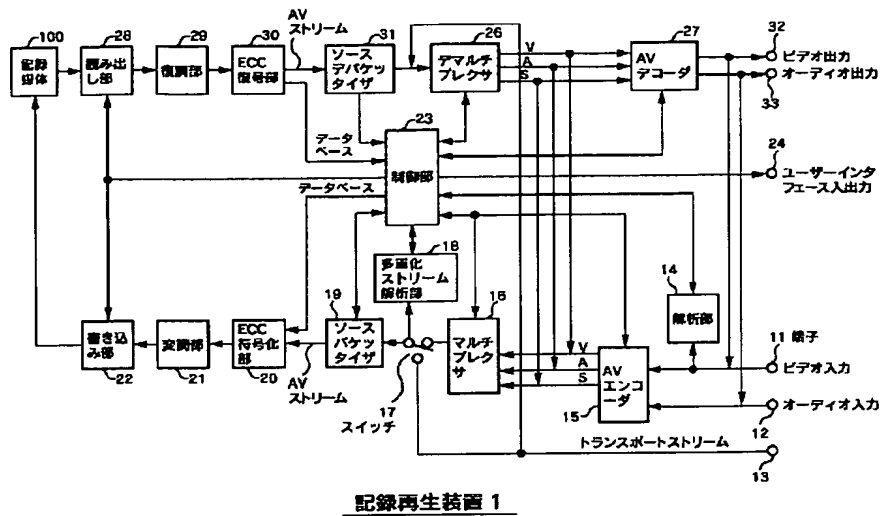
ResumeVolume のシンタクス

【図19】

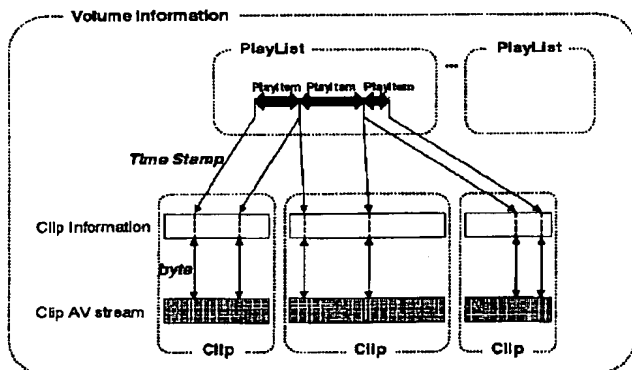
Value	Character coding
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

Character set value

【図 1】

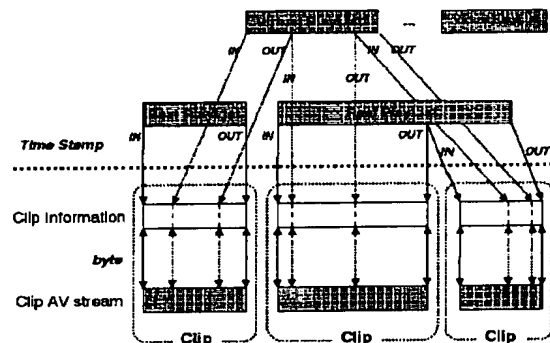


【図 2】



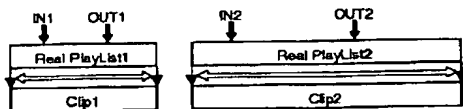
【図 6】

【図 3】

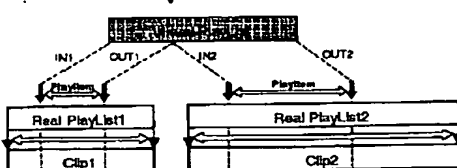


【図 7】

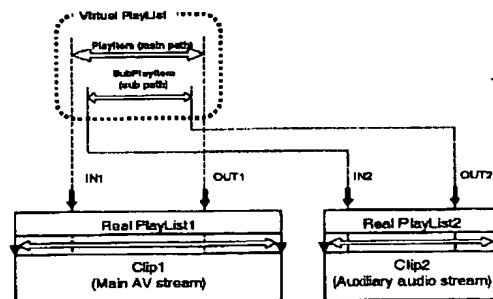
(A)



(B)



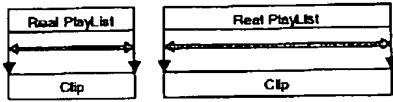
アセンブル編集の例



Virtual Playlist へのオーディオのアフレコの例

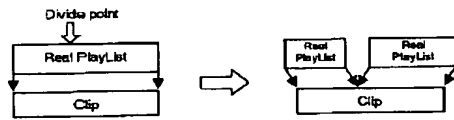
【図4】

(A)



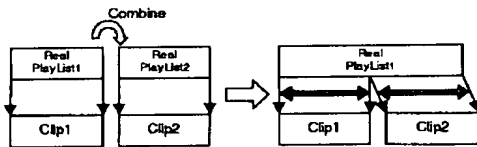
Real Playlistのクリエイトの例

(B)



Real Playlistのディバイドの例

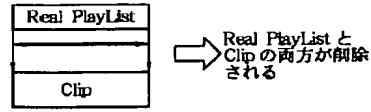
(C)



Real Playlistのコンバインの例

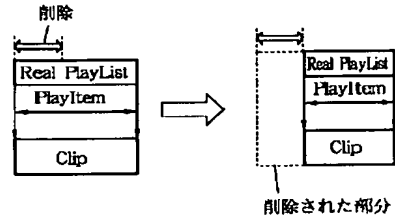
【図5】

(A)



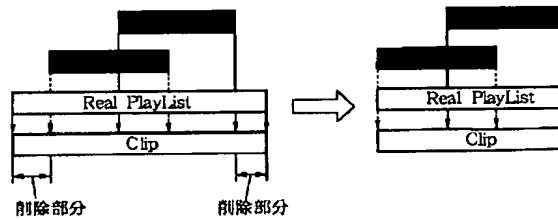
Real Playlist 全体のデリートの例

(B)



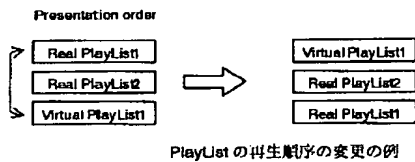
Real Playlist の部分的なデリートの例

(C)



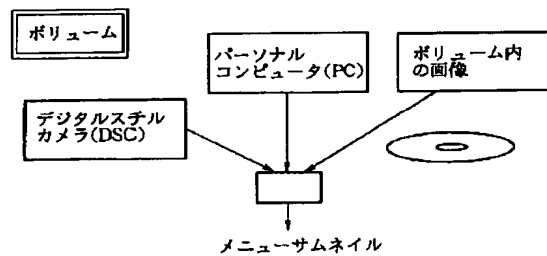
Real Playlist のミニマイズの例

【図8】

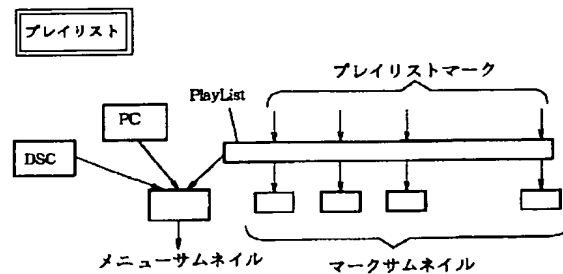


Playlistの再生順序の変更の例

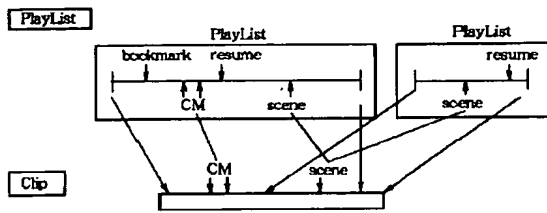
【図10】



【図11】

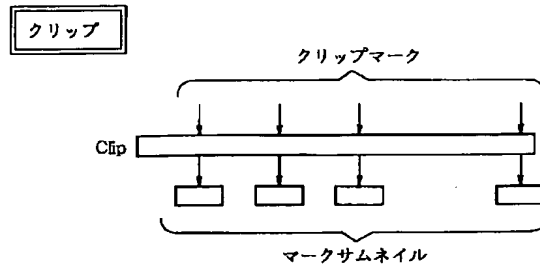


【図 9】

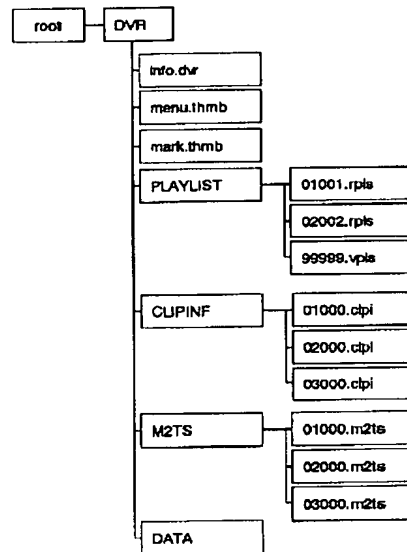


Playlist 上のマークと Clip 上のマーク

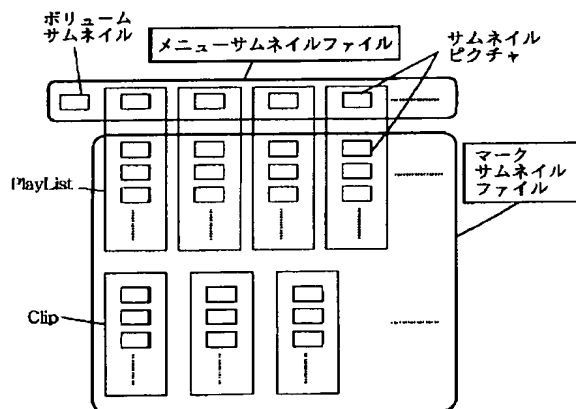
【図 12】



【図 14】



【図 13】



【図 15】

Syntax	No. of bits	Mnemonic
info.dvr {		
TableOfPlayLists_Start_address	32	uimbsf
MakerPrivateData_Start_address	32	uimbsf
reserved	192	bslbf
DVRVolume()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
MakerPrivateData()		
}		

Info.dvr のシンタクス

【図16】

Syntax	No. of bits	Mnemonics
DVRVolume() {		
version number	8*4	bslbf
length	32	uimbsf
ResumeVolume()		
UIAppInfoVolume()		
}		

DVR Volume のシンタクス

【図18】

Syntax	No. of bits	Mnemonics
UIAppInfoVolume() {		
character set	8	bslbf
name length	8	uimbsf
Volume name	8*256	bslbf
reserved	15	bslbf
Volume protect flag	1	bslbf
PIN	8*4	bslbf
ref thumbnail index	16	uimbsf
reserved for future use	256	bslbf
}		

UIAppInfoVolume のシンタクス

【図20】

Syntax	No. of bits	Mnemonics
TableOfPlayLists() {		
version number	8*4	bslbf
length	32	uimbsf
number of PlayLists	16	uimbsf
for (i=0; i<number of PlayLists; i++) {		
Playlist file name	8*10	bslbf
}		
}		

TableOfPlayLists のシンタクス

【図21】

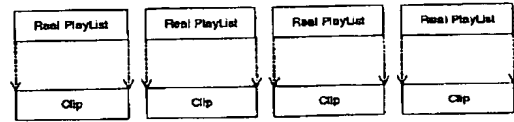
■ TableOfPlayLists ・ シンタクス (4.2.3.2 の別案)

Syntax	No. of bits	Mnemonics
TableOfPlayLists() {		
version number	8*4	bslbf
length	32	uimbsf
number of PlayLists	16	uimbsf
for (i=0; i<number of PlayLists; i++) {		
Playlist file name	8*10	bslbf
UIAppInfoPlaylist()		
}		
}		

TableOfPlayLists の別シンタクス

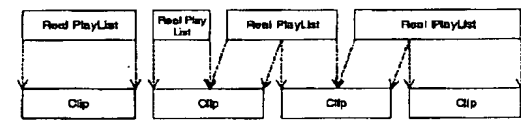
【図24】

(A)



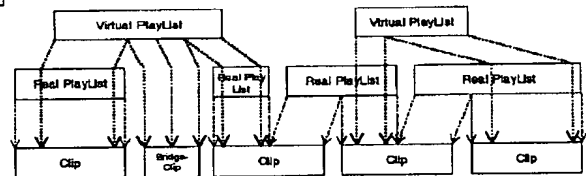
初めて AV ストリームが Clip として記録された時の Real Playlist の例

(B)



編集後の Real Playlist の例

(C)



Virtual Playlist の例

【図26】

Playlist type	Meaning
0	AV 記録のための Playlist この Playlist に参照されるすべての Clip は、一つ以上のビデオストリームを含まなければならない。
1	オーディオ記録のための Playlist この Playlist に参照されるすべての Clip は、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2 - 255	reserved

Playlist_type

【図41】

SubPath type	Meaning
0x00	Auxiliary audio stream path
0x01 - 0xff	reserved

SubPath_type

【 図 2 2 】

Syntax	No. bits	of	Minemonics
MakersPrivateData() {			
version number	8*4		bslbf
length	32		uimsbf
if (length != 0) {			
mpd blocks start address	32		uimsbf
number of maker entries	16		uimsbf
mpd block size	16		uimsbf
number of mpd blocks	16		uimsbf
reserved	16		bslbf
for (i=0; i<number of maker entries; i++) {			
maker ID	16		uimsbf
maker model code	16		uimsbf
start mpd block number	16		uimsbf
reserved	16		bslbf
mpd length	32		uimsbf
}			
stuffing bytes	8*2*L1		bslbf
for (i=0; i<number of mpd blocks; i++) {			
mpd_block	mpd_block_size*1024*8		
}			
}			

MakersPrivateData のシンタクス

【 図 2 3 】

Syntax	No. bits	of	Minemonics
xxxxx.rpls / yyyyy.vpls {			
PlaylistMark Start address	32		uimsbf
MakerPrivateData Start address	32		uimsbf
reserved	192		bslbf
Playlist() {			
for (i=0; i<N1; i++) {			
padding word	16		bslbf
}			
PlaylistMark() {			
for (i=0; i<N2; i++) {			
padding word	16		bslbf
}			
MakerPrivateData() {			
}			

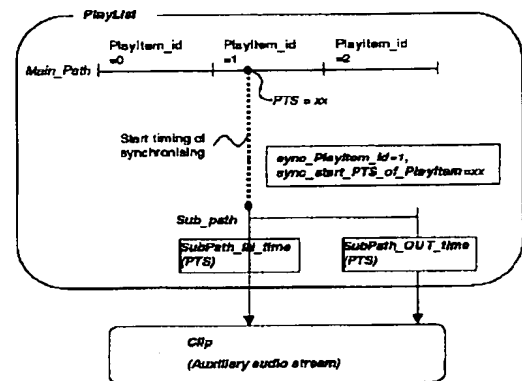
xxxxx.rpls と yyyyy.vpls のシンタクス

【 図 3 2 】

Syntax	No. bits	of	Minemonics
PlayItem() {			
Clip information file name	8*10		bslbf
reserved	24		bslbf
STC sequence Id	8		uimsbf
IN time	32		uimsbf
OUT time	32		uimsbf
reserved	14		bslbf
connection condition	2		bslbf
if (<Virtual PlayList>) {			
if (connection condition == '10') {			
BridgeSequenceInfo()			
}			
}			
}			

PlayItem のシンタクス

【 図 3 9 】



【図 25】

Syntax	No. of bits	Mnemonics
PlayList() {		
version_number	8*4	bslbf
length	32	uimsbf
PlayList_type	8	uimsbf
CPI_type	1	bslbf
reserved	7	bslbf
UIAppInfoPlayList() {		
number_of_PlayItems // main path	16	uimsbf
if (<Virtual PlayList>) {		
number_of_SubPlayItems // sub path	16	uimsbf
} else {		
reserved	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++) {		
PlayItem() // main path		
}		
if (<Virtual PlayList>) {		
if (CPI_type==0 && PlayList_type==0) {		
for (i = 0; i < number_of_SubPlayItems; i++)		
SubPlayItem() // sub path		
}		
}		
}		

PlayList のシンタックス

【図 27】

Syntax	No. of bits	Mnemonics
UIAppInfoPlayList2() {		
character_set	8	bslbf
name_length	8	uimsbf
PlayList_name	8*256	bslbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*8	bslbf
valid_period	4*8	bslbf
maker_id	16	uimsbf
maker_code	16	uimsbf
reserved	11	bslbf
playback_control_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
archive	2	bslbf
ref_thumbnail_index	16	uimsbf
reserved_for_future_use	256	bslbf
}		

UIAppInfoPlayList のシンタックス

【図 33】

CPI_type in the PlayList()	Semantics of IN_time
EP_map type	IN_time は、PlayItem の中で最初のプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	IN_time は、TU_map_time_axis 上の時刻でなければならない。かつ、IN_time は、time_unit の精度に丸めて表さなければならない。IN_time は、次に示す等式により計算される。
	$IN_time = TU_start_time \% 2^{32}$

IN_time

【図 47】

Clip_stream_type	meaning
0	Clip AV ストリーム
1	Bridge-Clip AV ストリーム
2 - 255	Reserved

Clip_stream_type

【図28】

(A)

write_protect_flag	Meaning
0b	その PlayList を自由に消去しても良い。
1b	write_protect_flag を除いてその PlayList の内容は、消去および変更されるべきではない。

write_protect_flag

(B)

is_played_flag	Meaning
0b	その PlayList は、記録されてから一度も再生されたことがない。
1b	PlayList は、記録されてから一度は再生された。

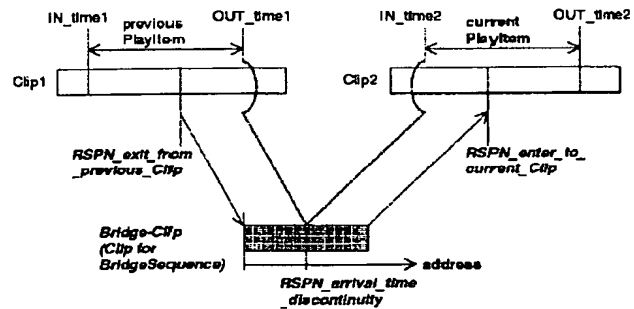
is_played_flag

(C)

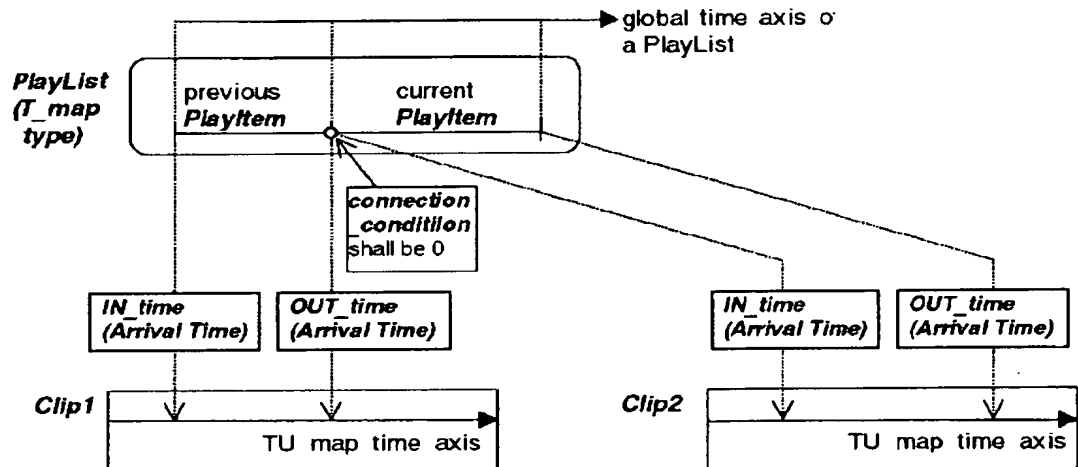
archive	Meaning
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

archive

【図37】

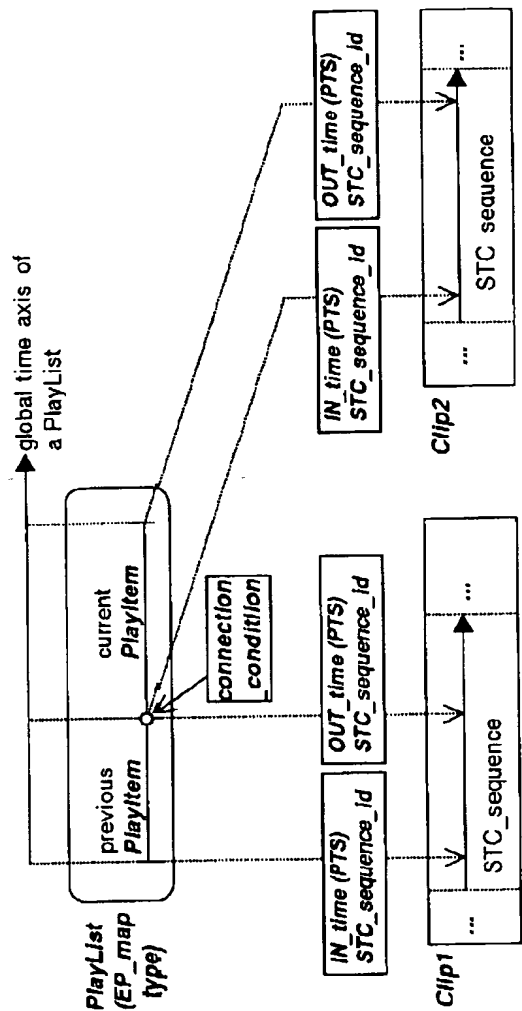


【図31】



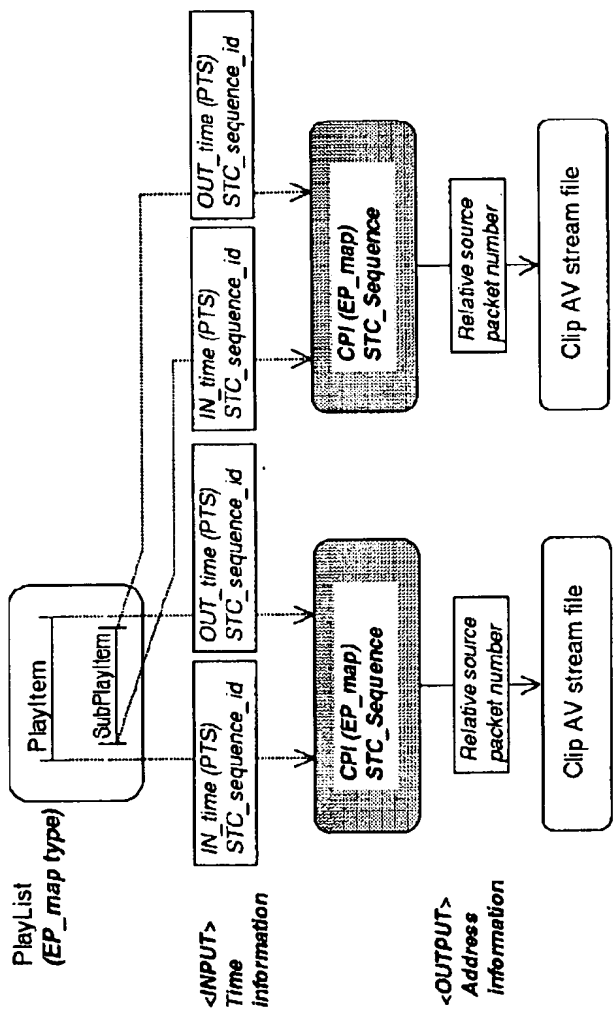
Playlist が TU_map type である時の例

【図29】

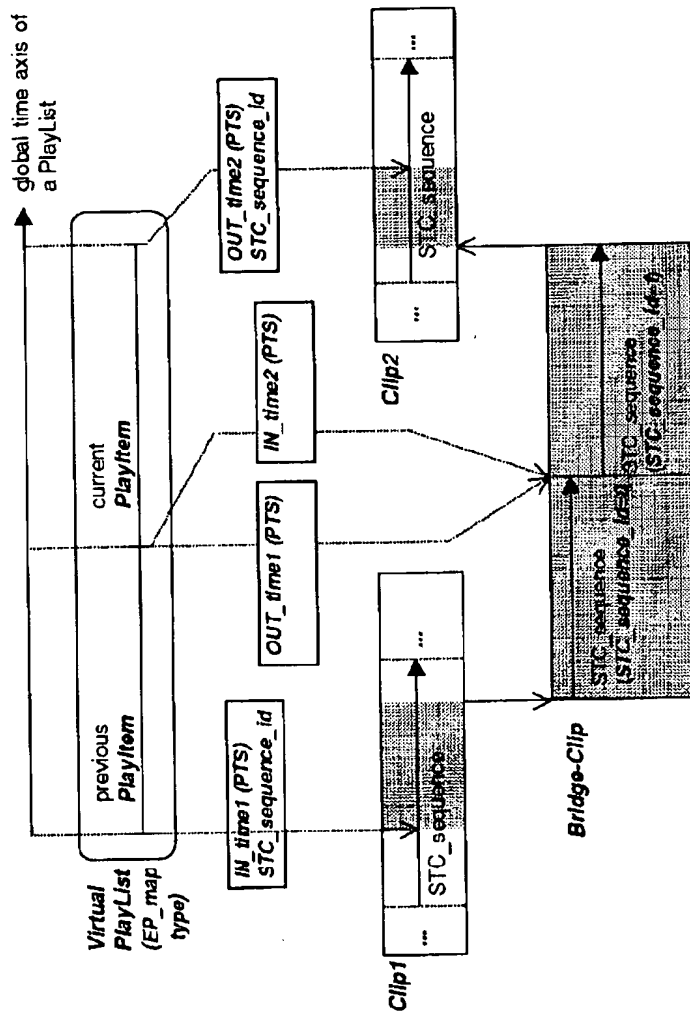


PlayList が EP_map type であり、かつ PlayItem が BridgeSequence を持たない時の例

【図63】

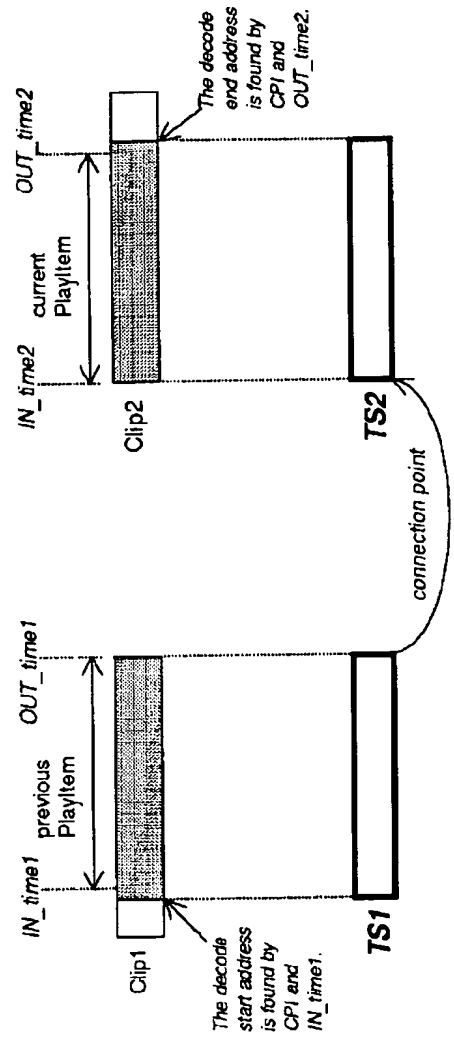


【 図 3 0 】



【 図 8 9 】

Playlist が EP_map type であり、かつ PlayItem が BridgeSequence を持つ時の例



【 図 3 4 】

CPI_type in the PlayList()	Semantics of OUT_time
EP_map type	<p>OUT_time は、次に示す等式によって計算される Presentation_end_TS の値の上位 32 ビットを示さなければならない。</p> $Presentation_end_TS = PTS_out + AU_duration$ <p>ここで、</p> <p>PTS_out は、PlayItem の中で最後のプレゼンテーションユニットに対応する 33 ビット長の PTS である。</p> <p>AU_duration は、最後のプレゼンテーションユニットの 90kHz 単位の表示期間である。</p>
TU_map type	<p>OUT_time は、TU_map_time_axis 上の時刻でなければならない。かつ、OUT_time は、time_unit の精度に丸めて表さなければならない。OUT_time は、次に示す等式により計算される。</p> $OUT_time = TU_start_time \% 2^{32}$

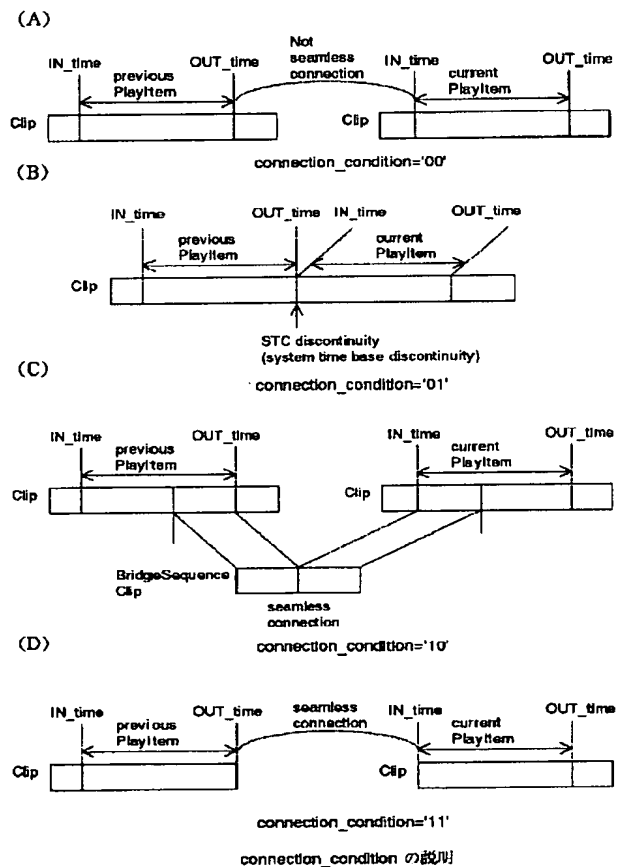
OUT_time

【 図 3 5 】

connection condition	meaning
00	<ul style="list-style-type: none"> 先行する PlayItem と現在の PlayItem の接続は、シームレス再生の保証がなされていない。 PlayList の CPI_type が TU_map type である場合、connection_condition は、この値をセットされなければならない。
01	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 先行する PlayItem と現在の PlayItem は、システムタイムベース (STC ベース) の不連続点があるために分割されていることを表す。
10	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 この状態は、Virtual PlayList に対してだけ許される。 先行する PlayItem と現在の PlayItem との接続は、シームレス再生の保証がなされている。 先行する PlayItem と現在の PlayItem は、BridgeSequence を使用して接続されており、DVR MPEG-2 トランスポートストリームは、後述する DVR-STD に従っていないなければならない。
11	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 先行する PlayItem と現在の PlayItem は、シームレス再生の保証がなされている。 先行する PlayItem と現在の PlayItem は、BridgeSequence を使用しないで接続されており、DVR MPEG-2 トランスポートストリームは、後述する DVR-STD に従っていないなければならない。

connection_condition

【 図 3 6 】



【 図 3 8 】

Syntax	No. of bits	of Mnemonics
BridgeSequenceInfo() {		
Bridge Clip information file name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimbsf
RSPN_enter_to_current_Clip	32	uimbsf
}		

BridgeSequenceInfo のシンタクス

【 図 4 0 】

Syntax	No. of bits	of Mnemonics
SubPlayItem() {		
Clip information file name	8*10	bslbf
SubPath type	8	bslbf
sync PlayItem id	8	uimbsf
sync start PTS of PlayItem	32	uimbsf
SubPath IN time	32	uimbsf
SubPath OUT time	32	uimbsf
}		

SubPlayItem のシンタクス

【 図 5 6 】

video format	Meaning
0	480i
1	576i
2	480p (including 840x480p format)
3	1080i
4	720p
5	1080p
6 - 254	reserved
255	No information

video_format

【 図 4 2 】

Syntax	No. of bits	of Mnemonics
PlayListMark() {		
version number	8*4	bslbf
length	32	uimbsf
number of PlayList marks	16	uimbsf
for(i=0; i < number of PlayList marks; i++) {		
reserved	8	bslbf
mark type	8	bslbf
mark time stamp	32	uimbsf
PlayItem id	8	uimbsf
reserved	24	uimbsf
character set	8	bslbf
name length	8	uimbsf
mark name	8*256	bslbf
ref_thumbnail_index	16	uimbsf
}		
}		

PlayListMark のシンタクス

【図43】

Mark type	Meaning	Comments
0x00	resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	book-mark	PlayListの再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークに使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログラムの最後まで、プレーヤはプログラムをスキップする。PlayListMark()において定義されるスキップマークポイントの数は、0または1でなければならない。
0x03 - 0xBF	reserved	
0x90 - 0xFF	reserved	Reserved for ClipMark()

mark_type

【図44】

CPI_type in the PlayList()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めて表さなければならない。mark_time_stamp は、次に示す等式により計算される。 $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

mark_time_stamp

【図62】

sampling frequency	Meaning
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

sampling_frequency

【図66】

CPI_type	Meaning
0	EP_map type
1	TU_map type

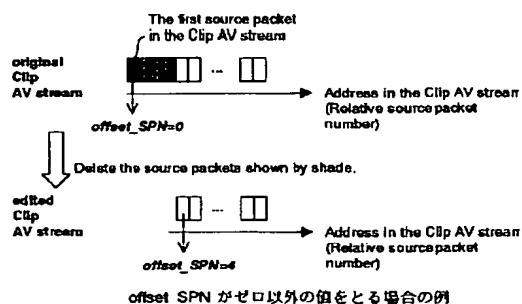
CPI_type の意味

【図46】

Syntax	No. bits	of	Mnemonics
ClipInfo()			
version number	8*4		bsbfi
length	32		uimabfi
Clip stream type	8		bsbfi
offset SPN	32		uimabfi
TS recording rate	24		uimabfi
reserved	8		bsbfi
record time and date	4*14		bsbfi
reserved	8		bsbfi
duration	4*8		bsbfi
reserved	7		bsbfi
time controlled flag	1		bsbfi
TS average rate	24		uimabfi
if (Clip stream type==1) // Bridge-Clip AV stream RSPN arrival time discontinuity	32		uimabfi
else reserved	32		bsbfi
reserved for system use	144		bsbfi
reserved	11		bsbfi
is format identifier valid	1		bsbfi
is original network ID valid	1		bsbfi
is transport stream ID valid	1		bsbfi
is service ID valid	1		bsbfi
is country code valid	1		bsbfi
format identifier	32		bsbfi
original network ID	16		uimabfi
transport stream ID	16		uimabfi
service ID	16		uimabfi
country code	24		bsbfi
stream format name	16*8		bsbfi
reserved for future use	256		bsbfi

ClipInfo のシンタクス

【図48】



【図57】

frame_rate	Meaning
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97...)
5	30
6	50
7	60 000/1001 (59.94...)
8	60
9 - 254	reserved
255	No information

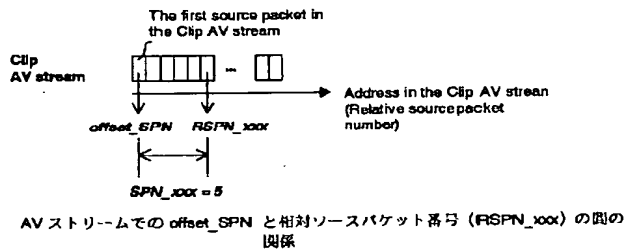
frame_rate

【図45】

Syntax	No. of bits	Mnemonic
zzzzz.clpi {		
STC_Info_Start_address	32	uimsbf
ProgramInfo_Start_address	32	uimsbf
CPI_Start_address	32	uimsbf
ClipMark_Start_address	32	uimsbf
MakerPrivateDate_Start_address	32	uimsbf
reserved	96	bslbf
ClipInfo()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
STC_Info()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
ProgramInfo()		
for (i=0; i<N3; i++) {		
padding_word	16	bslbf
}		
CPI()		
for (i=0; i<N4; i++) {		
padding_word	16	bslbf
}		
ClipMark()		
for (i=0; i<N5; i++) {		
padding_word	16	bslbf
}		
MakerPrivateDate()		
}		

zzzzz.clpi のシンタクス

【図49】

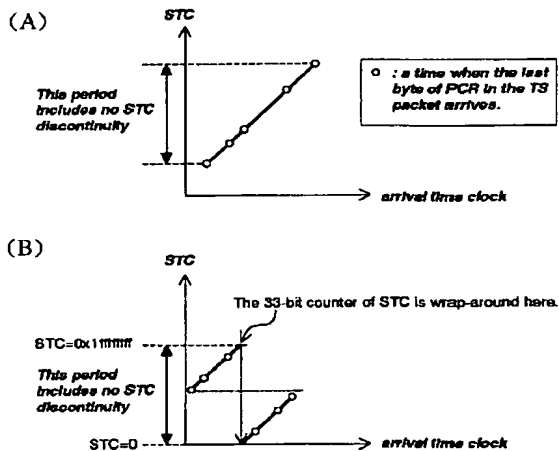


【図61】

audio_component_type	Meaning
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

audio_component_type

【図50】

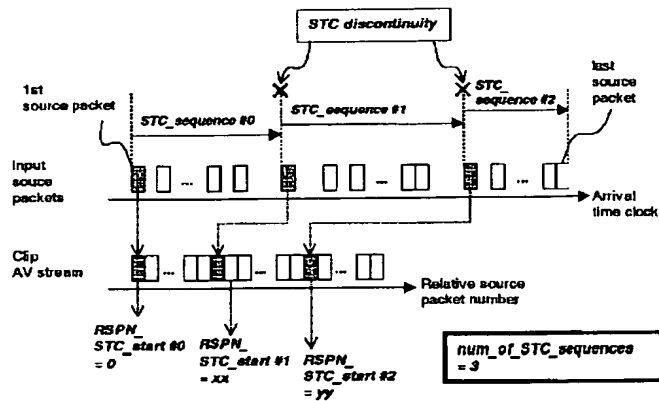


【図52】

Syntax	No. of bits	Mnemonic
STC_Info() {		
version number	8*4	bslbf
length	32	uimsbf
if (length != 0) {		
reserved	8	bslbf
num of STC sequences	8	uimsbf
for (STC_sequence_id=0; STC_sequence_id < num_of_STC_sequences; STC_sequence_id++) {		
reserved	32	bslbf
RSPN STC start	32	uimsbf
}		
}		

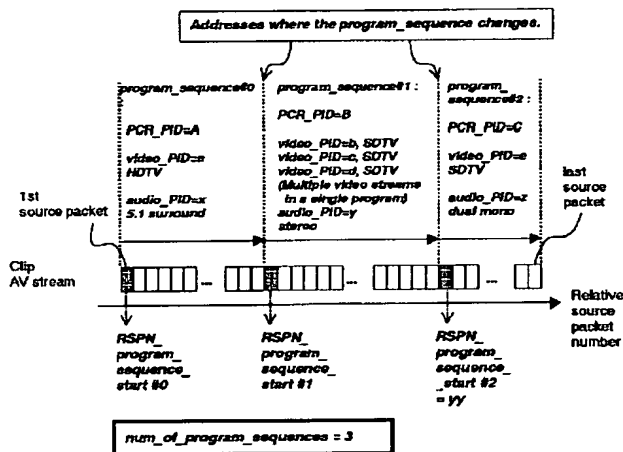
STC_Info のシンタクス

【図51】



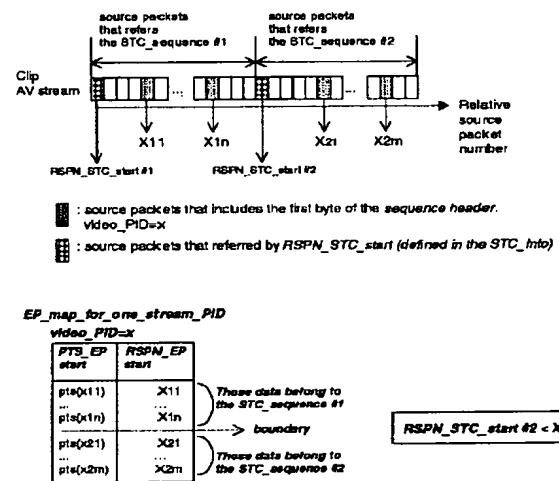
--STC_Info

【図53】



ProgramInfo の例

【図68】



【図55】

Syntax	No. bits	of	Mnemonics
VideoCodingInfo() {			
video format	8		uimsbf
frame rate	8		uimsbf
display aspect ratio	8		uimsbf
reserved	8		bsbf
}			

VideoCodingInfo のシンタクス

【 54】

Syntax	No. of bits	of Mnemonics
ProgramInfo {		
version number	8*4	bslbf
length	32	uimsbf
if (length != 0) {		
reserved	8	bslbf
number of program sequences	8	uimsbf
for (i=0; i<number of program sequences; i++){		
RSPN program sequence start	32	uimsbf
reserved	48	bslbf
PCR PID	16	bslbf
number of videos	8	uimsbf
number of audios	8	uimsbf
for (k=0; k<number of videos; k++){		
video stream PID	16	bslbf
VideoCodingInfo()		
}		
for (k=0; k<number of audios; k++){		
audio stream PID	16	bslbf
AudioCodingInfo()		
}		
}		
}		

ProgramInfo のシンタクス

【 59】

Syntax	No. of bits	of Mnemonics
AudioCodingInfo {		
audio coding	8	uimsbf
audio component type	8	uimsbf
sampling frequency	8	uimsbf
reserved	8	bslbf
}		

AudioCodingInfo のシンタクス

【 65】

Syntax	No. of bits	of Mnemonics
CPI {		
version number	8*4	bslbf
length	32	uimsbf
reserved	15	bslbf
CPI type	1	bslbf
if (CPI type == 0)		
EP_map()		
else		
TU_map()		
}		

CPI のシンタクス

【 80】

Thumbnail picture format	Meaning
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

thumbnail_picture_format

【 58】

display aspect ratio	Meaning
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ratio
4-254	reserved
255	No information

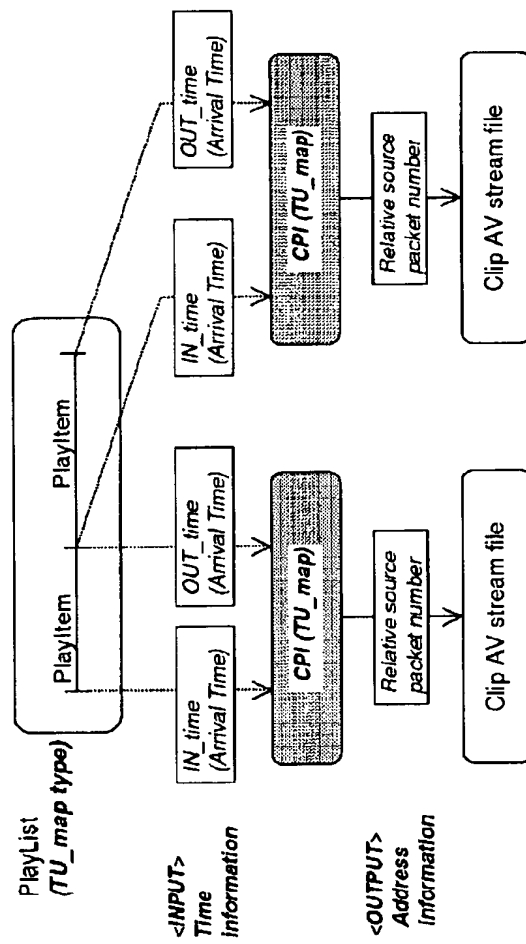
display_aspect_ratio

【 60】

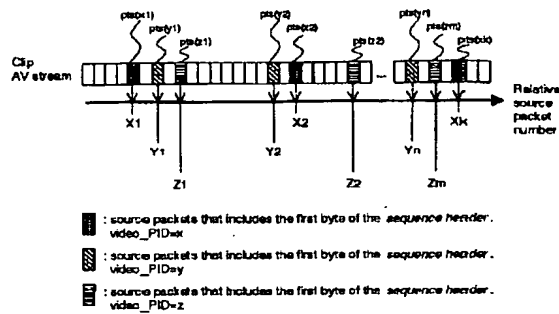
audio coding	Meaning
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

audio_coding

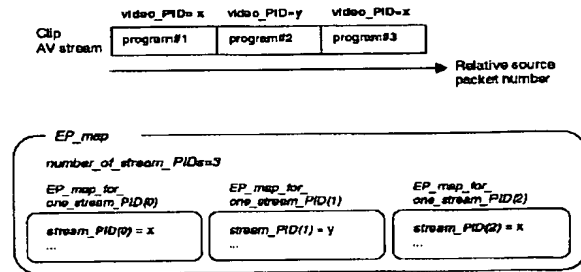
【 図 64 】



【図67】



【図69】



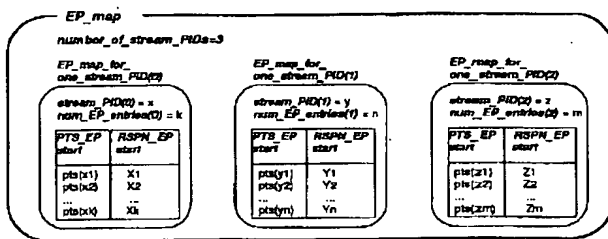
【図76】

Mark type	Meaning	Comments
0x00 - 0x8F	reserved	Reserved for PlaylistMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点。
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点。
0x92	Scene-start mark	シーンチェンジポイントを示すマーク。
0x93 - 0xFF	reserved	

mark_type

ビデオの EP_map の例

【図70】



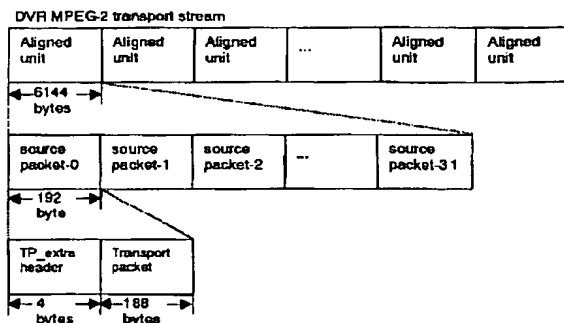
【図71】

Syntax	No. of bits	Mnemonics
EP_map()		
reserved	12	bsbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0; k<number_of_stream_PIDs; k++){		
stream_PID(k)	16	bsbf
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_Start_addresses(k)	32	uimsbf
}		
for (i=0; i<X; i++){		
padding_word	16	bsbf
}		
for (k=0; k<number_of_stream_PIDs; k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (l=0; l<Y; l++){		
padding_word	16	bsbf
}		
}		
}		

EP_type	Meaning
0	video
1	audio
2 - 15	reserved

EP_type Values

【図82】



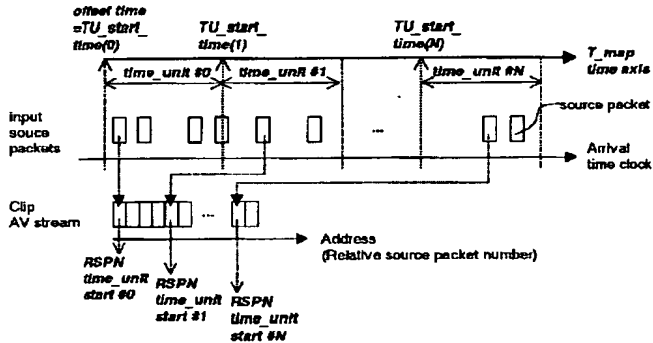
DVR MPEG-2 トランスポートストリームの構造

【図72】

Syntax	No. of bits	Mnemonics
EP_map_for_one_stream_PID(N){		
for (l=0; l<N; l++) {		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

EP_map_for_one_stream_PID のシンタクス

【図73】



【図74】

Syntax	No. of bits	Mnemonics
TU_map0{		
offset time	32	bsbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0; k<number_of_time_unit_entries; k++)		
RSPN_time_unit_start	32	uimsbf
}		

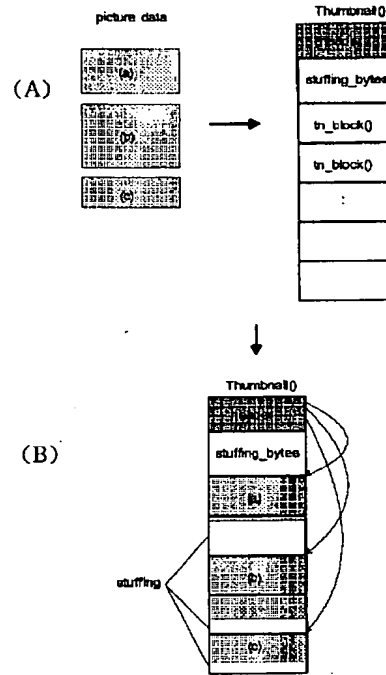
TU_map のシンタクス

【図77】

CPI_type In the CPI()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めて表さなければならない。mark_time_stamp は、次に示す等式により計算される。 $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

mark_type_stamp

【図81】

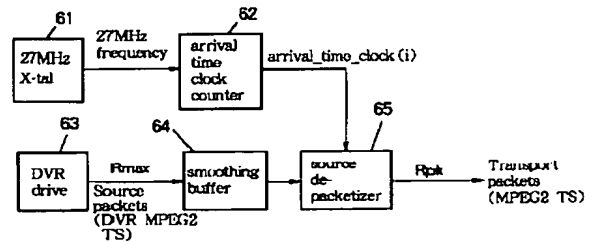


【図87】

copy_permission indicator	meaning
00	copy free
01	no more copy
10	copy once
11	copy prohibited

copy permission indicator table

【図84】



DVR MPEG-2 トランスポートストリームのプレーヤモデル

【図75】

Syntax	No. of bits	Mnemonics
ClipMark() {		
version number	8*4	bslbf
length	32	uimbsf
number of Clip marks	16	uimbsf
for(i=0; i < number of Clip marks; i++) {		
reserved	8	bslbf
mark_type	8	bslbf
mark_time_stamp	32	uimbsf
STC sequence id	8	uimbsf
reserved	24	bslbf
character set	8	bslbf
name length	8	uimbsf
mark name	8*256	bslbf
ref_thumbnail_index	16	uimbsf
}		
}		

ClipMark のシンタクス

【図78】

Syntax	No. of bits	Mnemonics
menu.thmb / mark.thmb {		
reserved	256	bslbf
Thumbnail()		
for(i=0; i < N1; i++)		
padding_word	16	bslbf
}		

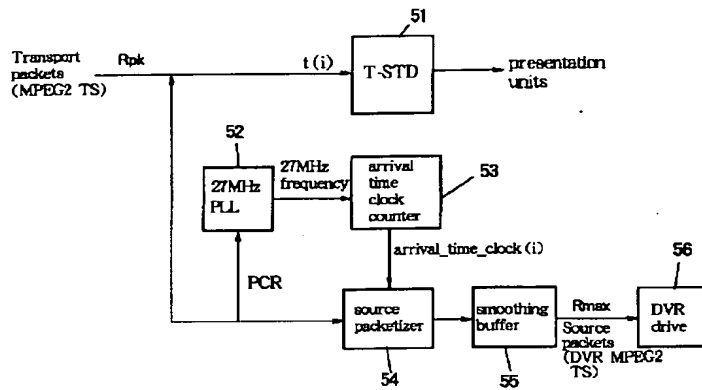
menu.thmb と mark.thmb のシンタクス

【図79】

Syntax	Bits	Mnemonics
Thumbnail() {		
version number	8*4	char
length	32	uimbsf
if (length != 0) {		
tn_blocks_start_address	32	bslbf
number of thumbnails	16	uimbsf
tn_block_size	16	uimbsf
number of tn blocks	16	uimbsf
reserved	16	bslbf
for(i = 0; i < number of thumbnails; i++) {		
thumbnail_index	16	uimbsf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimbsf
start tn_block number	16	uimbsf
x_picture_length	16	uimbsf
y_picture_length	16	uimbsf
reserved	16	uimbsf
}		
}		
stuffing_bytes	8*2*L1	bslbf
for(k = 0; k < number of tn blocks; k++) {		
tn_block	tn_block_size*1024*8	
}		
}		

Thumbnail のシンタクス

【図83】



DVR MPEG-2 トランスポートストリームのレコーダモデル

【図85】

Syntax	No. bits	of	Mnemonics
source_packet () {			
TP_extra_header()			
transport_packet()			
}			

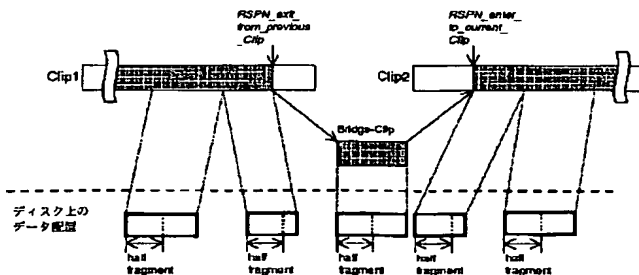
source packet

【図86】

Syntax	No. bits	of	Mnemonics
TP_extra_header() {			
copy_permission_indicator	2		uimsbf
arrival_time_stamp	30		uimsbf
}			

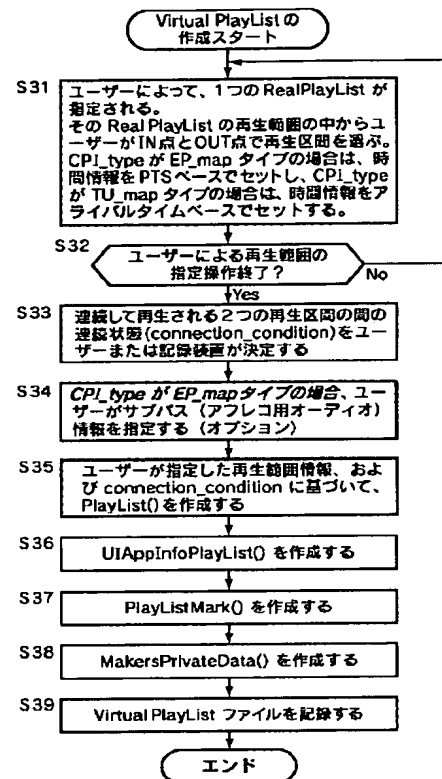
TP_extra_header

【図94】



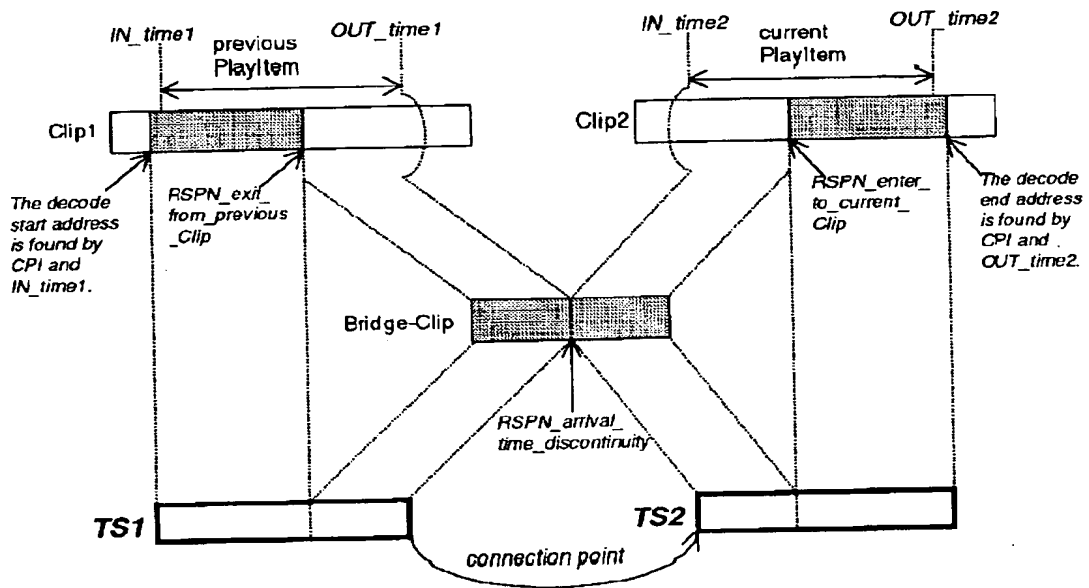
BridgeSequence を使用してシームレス接続をする場合の、データアロケーションの例

【図103】

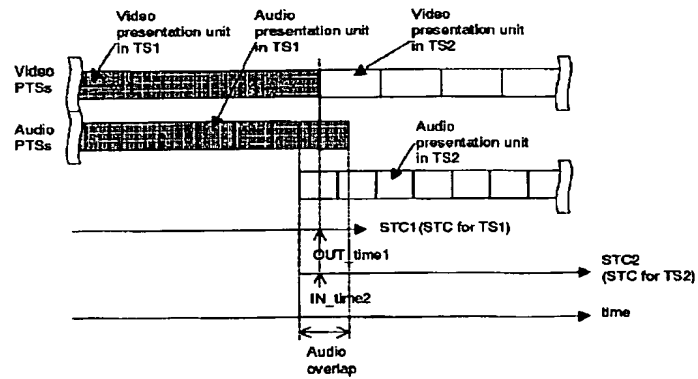


Virtual Playlist の作成方法を説明するフローチャート

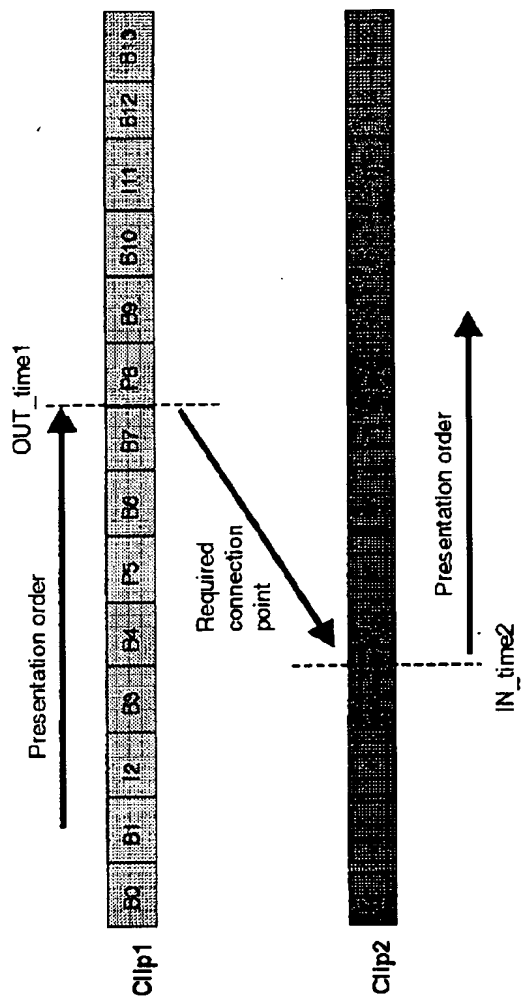
【図88】



【図93】

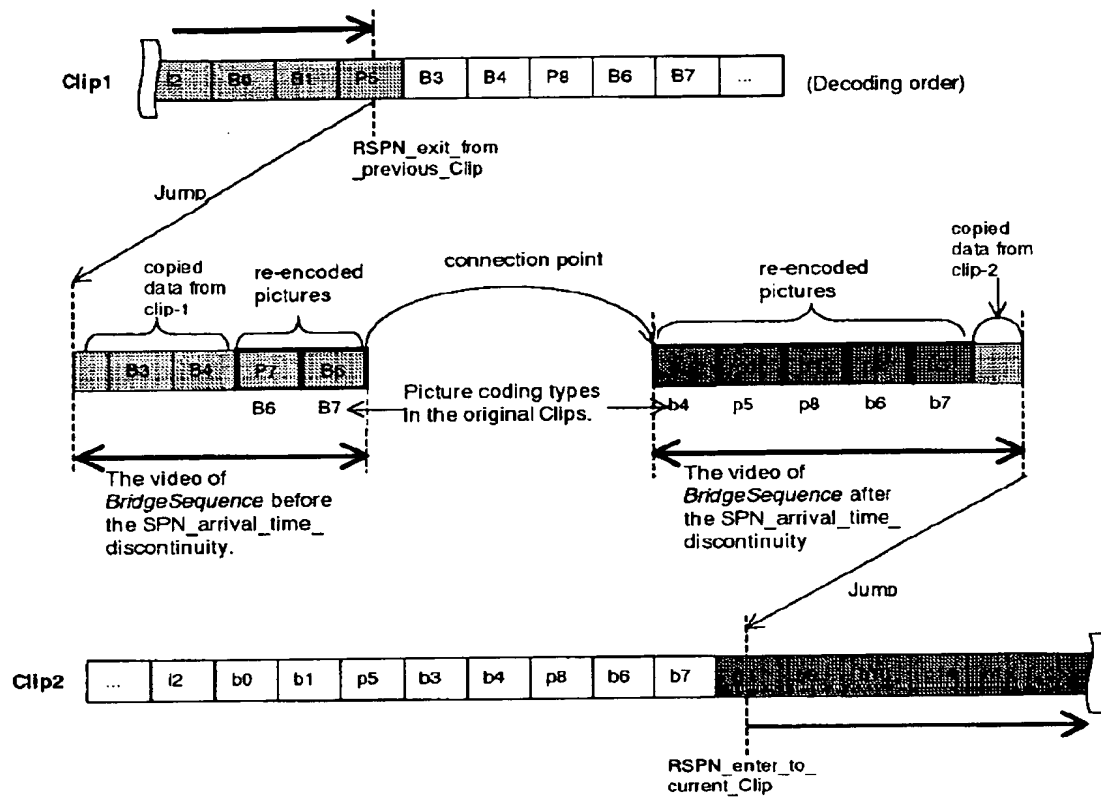


【図90】



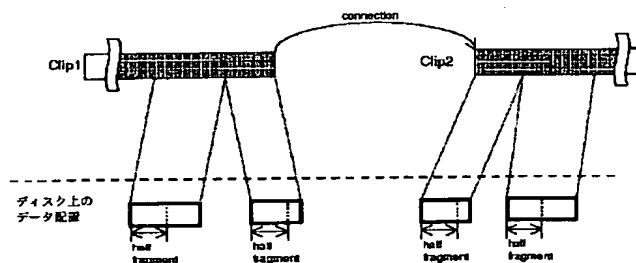
ビクチャの表示順序で示すシームレス接続の例

【 図 9 1 】



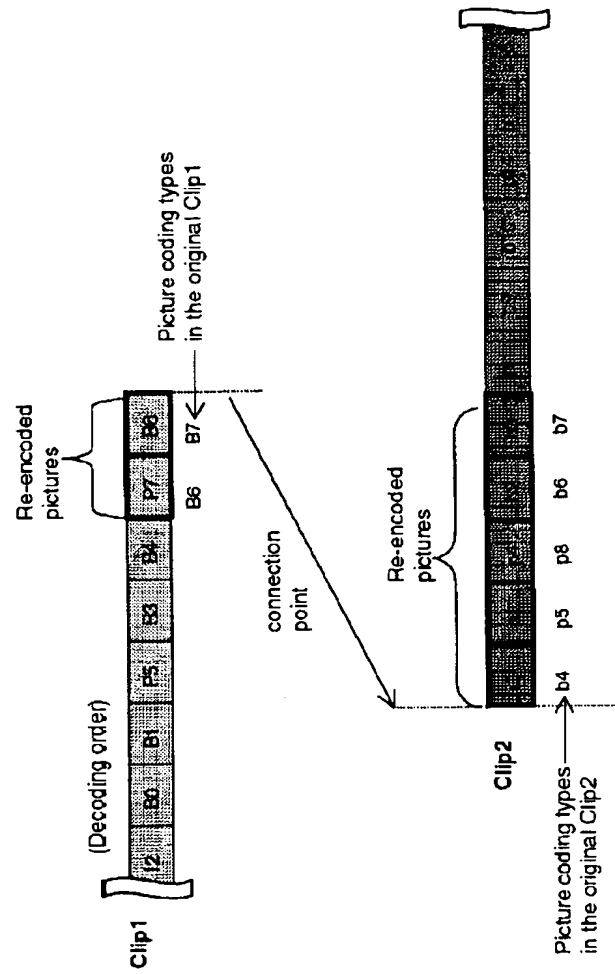
BridgeSequence を使用してシームレス接続を実現する例 1

【 図 9 5 】



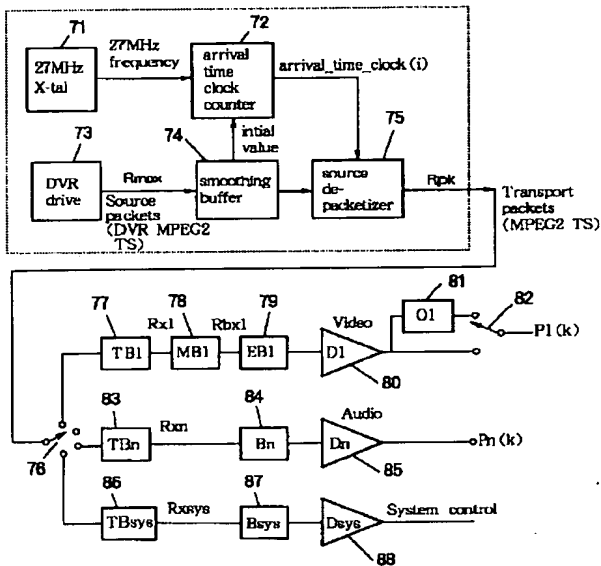
BridgeSequence を使用しないでシームレス接続をする場合の、データアロケーションの例

【 9 2 】



BridgeSequence を使用しないでシームレス接続を実現する例 2

【 96 】

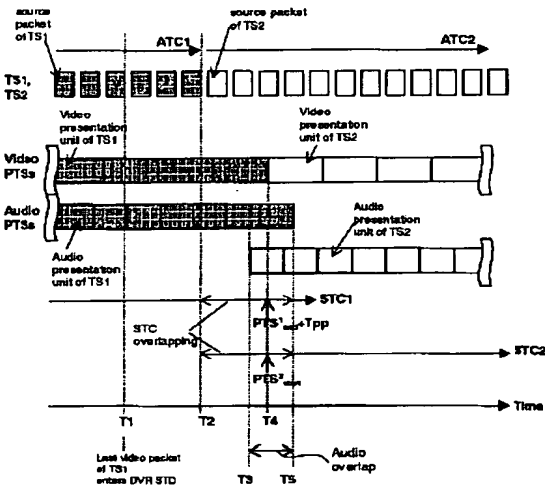


【 100 】

Syntax	No. of bits	Mnemonic
Playlist() {		
length	32	uimsbf
reserved_for_word_align	15	bslbf
CPI_type	1	bslbf
number_of_PlayItems	16	uimsbf
if (<Virtual-Playlist> && CPI_type==0) {		
number_of_SubPlayItems	16	uimsbf
} else {		
reserved_for_word_align	16	bslbf
}		
for (PlayItem_id=0;		
PlayItem_id<number_of_PlayItems;		
PlayItem_id++) {		
PlayItem()		
}		
if (<Virtual-Playlist> && CPI_type==0) {		
for (i=0; i<number_of_SubPlayItems; i++) {		
SubPlayItem()		
}		
}		
}		

図98の Playlist ファイルの中の Playlist() のシンタックス

【 97 】



ある AV ストリーム(TS1)からそれにシームレスに接続された次の AV ストリーム(TS2)へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャート

【図98】

Syntax	No. of bits	Mnemonic
xxxxx.rpls / yyyyy.vpls {		
version_number	8*4	bslbf
PlayList_start_address	32	uimsbf
PlayListMark_start_address	32	uimsbf
MakersPrivateData_start_address	32	uimsbf
reserved_for_future_use	160	bslbf
UIAppInfoPlayList()		
for (i=0; i<N1; i++) {		
padding_word	16	bslbf
}		
PlayListst()		
for (i=0; i<N2; i++) {		
padding_word	16	bslbf
}		
PlayListstMark()		
for (i=0; i<N3; i++) {		
padding_word	16	bslbf
}		
MakersPrivateData()		
for (i=0; i<N4; i++) {		
padding_word	16	bslbf
}		
}		

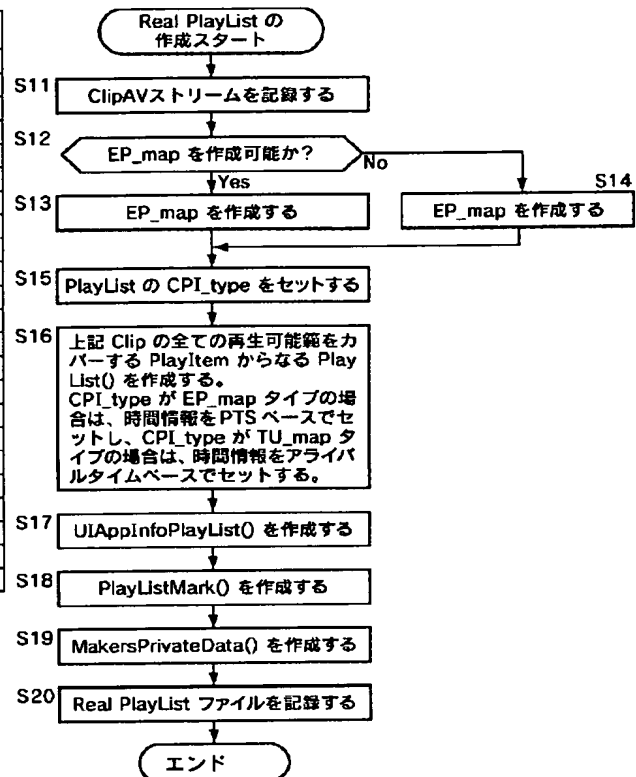
PlayList ファイルのシンタックスの別例

【図99】

Syntax	No. of bits	Mnemonic
UIAppInfoPlayList() {		
length	32	uimsbf
PlayList_service_type		
PlayList_character_set	8	uimsbf
reserved_for_word_align	3	bslbf
playback_control_flag	1	uimsbf
write_protect_flag	1	uimsbf
is_played_flag	1	uimsbf
archive	2	uimsbf
record_time_and_date	4*14	bslbf
duration	4*6	bslbf
maker ID	16	uimsbf
maker model code	16	uimsbf
ref_thumbnail_index	16	uimsbf
reserved	7	bslbf
rp_info_valid_flag	1	uimsbf
rp_ref_to_PlayItem_id	16	uimsbf
rp_time_stamp	32	uimsbf
channel_number	16	uimsbf
reserved_for_word_align	8	bslbf
channel_name_length	8	uimsbf
channel_name	8*20	bslbf
PlayList_name_length	8	uimsbf
PlayList_name	8*255	bslbf
PlayList_detail_length	16	uimsbf
PlayList_detail	8*1200	bslbf
}		

図98の PlayList ファイルの中の UIAppInfoPlayList のシンタックス

【図102】



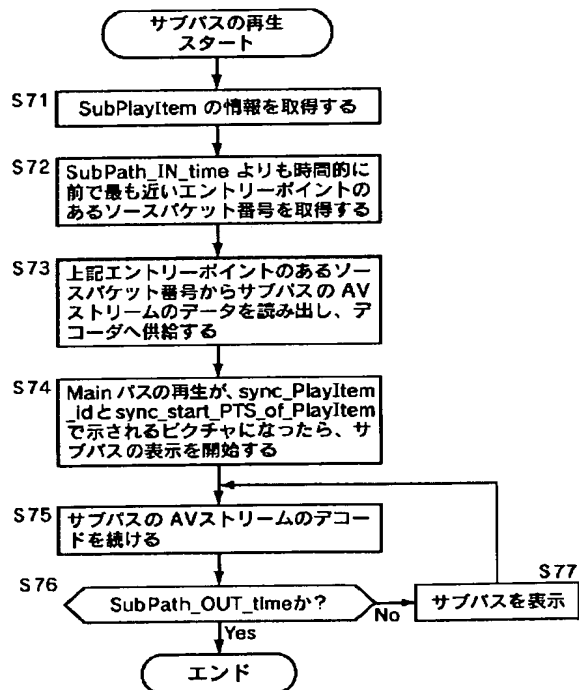
Real PlayList の作成方法を説明するフローチャート

【図101】

Syntax	No. of bits	Mnemonic
SubPlayItem() {		
length	16	uimsbf
Clip_information_file_name	8*10	bslbf
SubPath_type	8	bslbf
STC_sequence_id	8	uimsbf
SubPath_IN_time	32	uimsbf
SubPath_OUT_time	32	uimsbf
sync_PlayItem_id	16	uimsbf
sync_start_PTS_of_PlayItem	32	uimsbf
}		

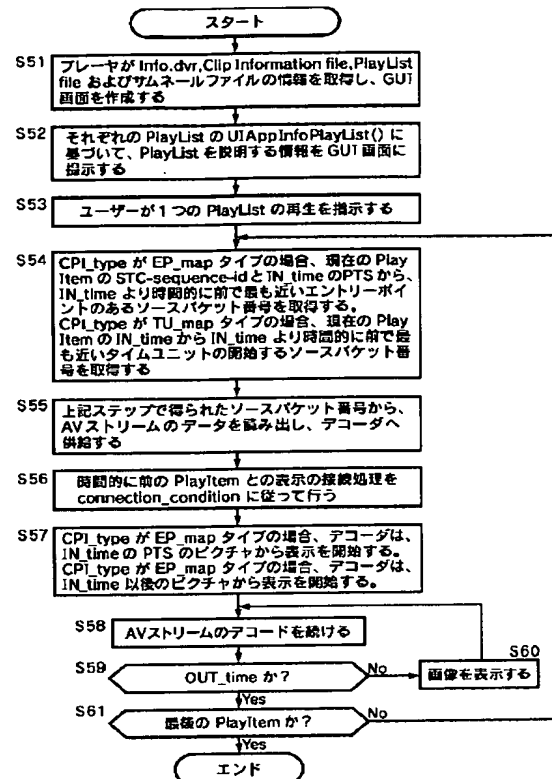
SubPlayItem のシンタックスの別例

【図105】



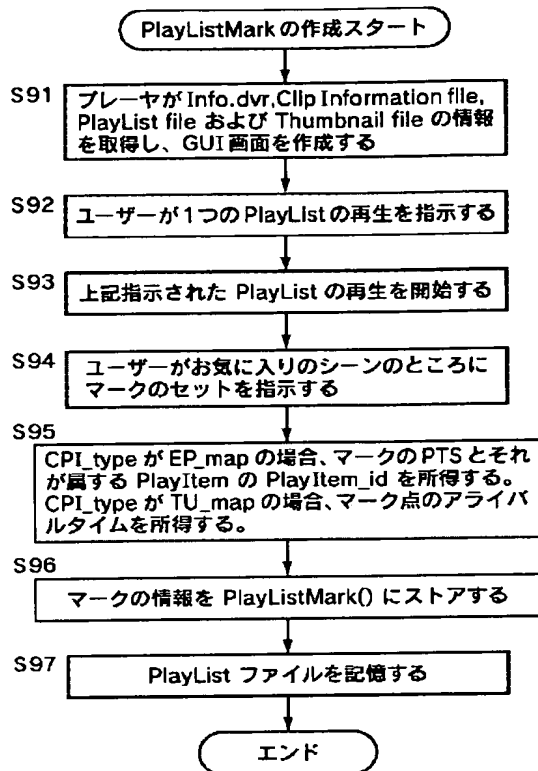
PlayList の Sub パスの再生方法を説明するフローチャート

【図104】



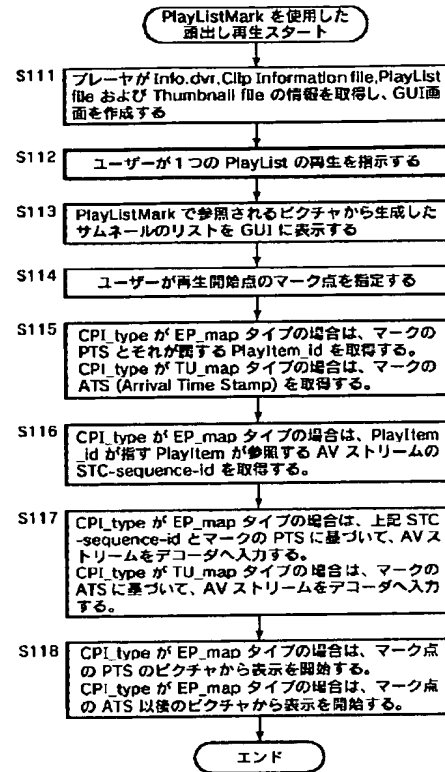
PlayList の再生方法を説明するフローチャート

【図106】



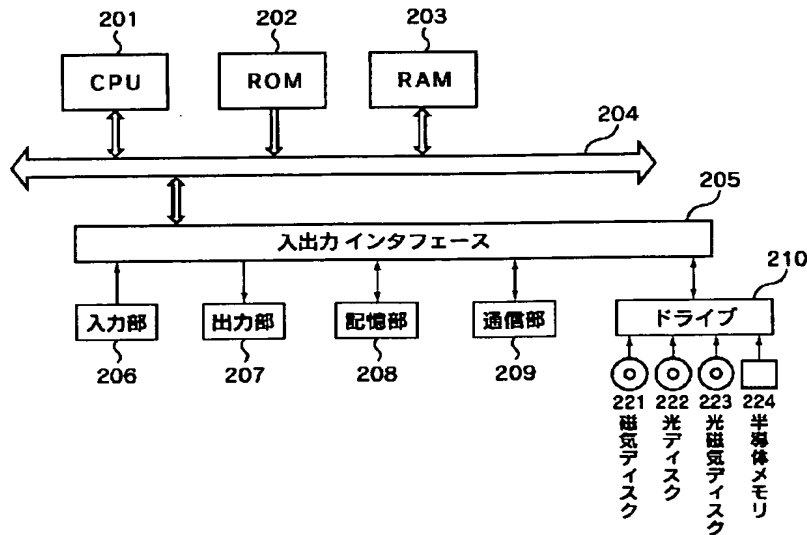
PlaylistMark の作成方法を説明するフローチャート

【図107】



PlaylistMark を使用した頭出しの再生方法を説明するフローチャート

【図108】



フロントページの続き

F ターム(参考) 5C052 AA02 AC05 AC08 DD04
5C053 FA14 FA20 FA23 FA29 GA11
GB05 GB06 GB09 GB38 HA24
HA29 JA24 KA08 KA24 KA26